STD 7000

7712

512K Dynamic RAM Card

USER'S MANUAL

FOREWORD

This manual explains how to use the Pro-Log 7712 512K Dynamic RAM Card. It is structured to reflect the answers to basic questions you might ask yourself about the 7712. We welcome your suggestions on how we can improve our instructions.

The 7712 is part of the Pro-Log Series 7000 STD BUS hardware. Our products are modular, and designed and built with second-sourced parts that are industry standards. They provide industrial managers with the means of using their own people to control the design, production, and maintenance of the products that use STD BUS hardware.

Pro-Log supports its products with thorough and complete documentation. Also, we teach courses on how to design with and use microprocessors and the STD BUS products.

You may find the following Pro-Log documents useful in your work: the Microprocessor User's Guide and the Series 7000 STD BUS Technical Manual. If you would like a copy of these documents, please write to us on your company letterhead.

CONTENTS

	Page
Foreword	ii
Figures	v
Section 1. Purpose and Main Features	1-1
Purpose	1-1
Features	1-1
Functional Description	1-4
Section 2. Installation and Specifications	2-1
Introduction	2-1
Installation	2-1
Memory	2-2
Cabling	2-2
Card Layout and Jumpers	2-2
I/O Address (W1, W7)	2-2
Memory Address (W5, W9)	2-6
Block Mode (W3, W4, W5)	2-7
Memory Parts (W6)	2-7
Processor Type (W8)	2-7
Parity Error Detection (W2)	2-7
Specifications	2-8
Section 3. Operation and Programming	3-1
Introduction	3-1
I/O Port Addressing	3-1
Memory Management	3-4
16-Bit Systems - 8088	3-4
8-Bit Systems - 8085	3-7
Parity Error Detection	3-15
Refresh	3-18
DMA Processor Compatibility	3-19
Section 4. Operating Software	4-1
Introduction	4-1
Memory Addresses	4-1
I/O Port Addresses	4-1
Jumper Configurations	4-1
Subroutines Provided	4-2
Subroutine Specifications	4-3
(DRAM TEST)	4-3
(INIT PAGE)	4-5
(COMP PAGE)	4-6
(INIT DATA)	4-8
(COMP DATA)	4-9
(WALKING ONE)	4-11
(OUT BLOCK)	4-13

Page

Section 5. Maintenance **5-1**

Reference Drawings 5-1
Return for Repair/Warranty 5-6

Figures

Figure		Page
1-1	7712 Dynamic RAM Card	1-2
1-2	Block Diagram	1-3
2-1	Installation	2-1
2-2	W7 I/O Address Jumper	2-3
2-3	Card Layout and Jumper Descriptions	2-4
2-4	Memory Address Mapping	2-6
2-5	Connector J1	2-8
2-6	Electrical and Environmental Specifications	2-8
2-7	Processor Read Timing	2-9
2-8	DMA Read Timing	2-10
2-9	Write Timing	2-11
2-10	Forced Refresh Timing	2-12
2-11	STD Edge Connector Pin List	2-13
3-1	16-Bit I/O Port Addressing (Lower 32K)	3-2
3-2	16-Bit I/O Port Addressing (Upper 32K)	3-2
3-3	10-Bit I/O Port Addresing	3-3
3-4	Memory in 16-Bit Systems	3-5
3-5	8088 Segment Mode Register	3-6
3-6	8085 32K Block Mode Register	3-7
3-7	Pool of Memory Blocks	3-8
3-8	Basic Arrangement 1	3-8
3-9	Basic Arrangement 2	3-9
3-10	Basic Arrangements 1 and 2	3-10
3-11	8085 64K Block Mode Register	3-11
3-12	Basic Arrangement 3	3-12
3-13	8085 64K Block Mode	3-13
3-14	8085 64K Block Enable	3-14
4-1	List of Subroutines	4-2
4-2	(DRAM TEST) Flowchart	4-4
4-3	(INIT PAGE) Flowchart	4-5
4-4	(COMP PAGE) Flowchart	4-7
4-5	(INIT DATA) Flowchart	4-8
4-6	(COMP DATA) Flowchart	4-10
4-7	(WALKING ONE) Flowchart	4-12
4-8	(OUT BLOCK) Flowchart	4-13
4-9	Software Routines	4-14
5-1	7712 Schematic	5-2
5-2	7712 Assembly Drawing	5-5

PURPOSE

The 7712 dynamic RAM card offers the economy of a large dynamic memory array with the performance benefits associated with static memory. The 7712 memory array is designed with sockets for the dynamic RAMs and is available in four options: 64K and 128K bytes using 64Kx1 DRAMs or 256K and 512K bytes using 256Kx1 DRAMs. The 7712 is compatible with the 8088 (5 MHz), 8085 (6.25 MHz), and DMA controller cards such as the Pro-Log 7388.

The 7712 provides high density semiconductor memory for the STD BUS. It greatly reduces system cost by reducing the number of cards needed to implement large memory schemes. The 7712 improves the efficiency of disk-based systems by providing enough RAM to keep frequently used files on line, and features parity error detection. The memory management logic of the 7712 provides for the efficient use of 16-bit 8088 microprocessor memory while offering 8085 compatibility. The 7712 is suitable for a number of memory management techniques such as allocating banks of memory to individual time-sharing operators or running unrelated programs on an interleaved basis.

The advanced dynamic RAM controller design eliminates all hardware and software overhead due to refreshing DRAMs. It automatically refreshes onboard RAM without wait states during all modes of processor and DMA operations. The static-like appearance permits hardware single-stepping of programs, continuous DMA access, and maintains data during resets.

FEATURES

- Compatible with 8085 (at 6.25 MHz), 8088 (at 5 MHz), and most 16- or 20-bit address DMA controllers
- Refresh transparency from power-up to power-down
- Four versions: 7712-512 (512K), 7712-256 (256K), 7712-128 (128K), and 7712-64 (64K).
- Software controlled memory bank selection in 64K or 32K byte blocks
- 8-, 10-, and partial 16-bit I/O decoding options
- Upper address latch for 20-bit DMA use
- Parity error detection circuit provides several bus and user options
- Uses industry standard 64K-bit or 256K-bit dynamic RAM
- Allows expansion to one megabyte with the 8088 processor and expansion to 512K bytes with the 8085 processor
- Single +5V operation
- Fully STD BUS buffered

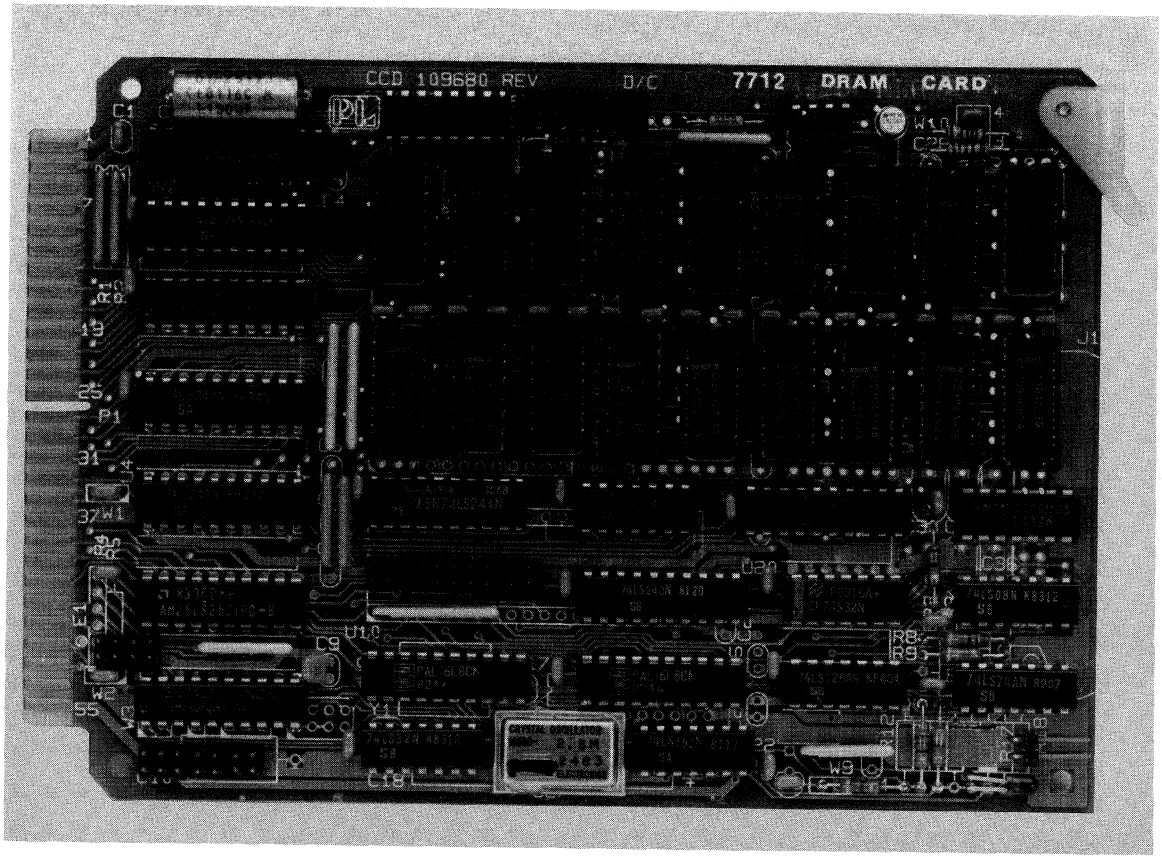


Figure 1-1. 7712 Dynamic RAM Card

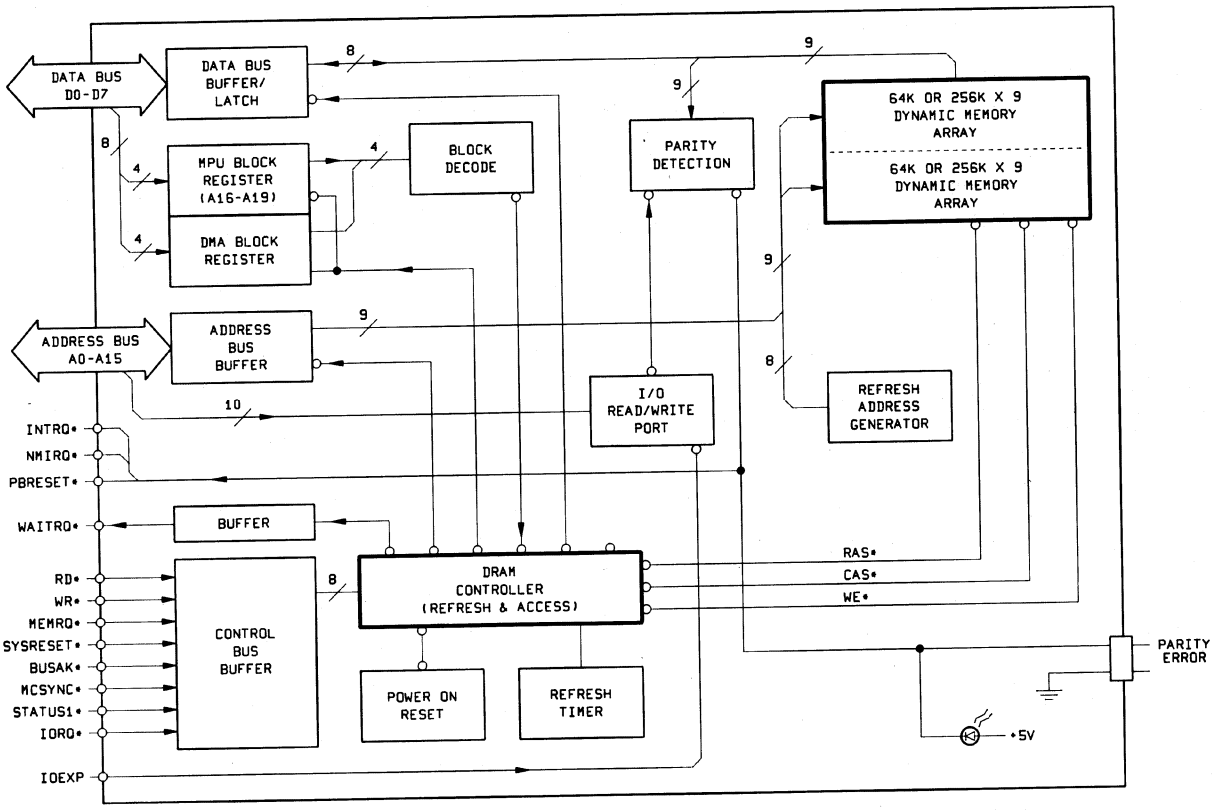


Figure 1-2. Block Diagram (7712)

FUNCTIONAL DESCRIPTION

The functional description defines each block in the 7712 block diagram (figure 1-2 on the opposite page). Together the diagram and functional description are designed to familiarize the user with terms used throughout the manual.

DATA BUS BUFFER/LATCH — provides buffering for the data bus, and latches data read from the DRAM memory. This allows static operation during extended WAIT states.

MPU BLOCK REGISTER — demultiplexes A16-A19 from the data bus in the 8088 mode or selects memory blocks in the 8085 modes.

DMA REGISTER — selects one of 16 memory blocks during DMA or selects the lower memory blocks in the 8085-64K mode.

ADDRESS BUS BUFFER — provides buffering and selection of memory.

REFRESH ADDRESS GENERATOR — provides memory row address during refresh.

CONTROL BUS BUFFER — provides the control signal interface between the STD BUS and the card.

PARITY DETECTION — generates parity for each byte written to memory and indicates parity status for each byte read from memory.

BLOCK DECODE — enables blocks of memory on 64K boundaries. In the 8088 mode, the memory is directly addressed. In the 8085 mode, 64K blocks are enabled, then one block is software selected and mapped into a 32K address space.

I/O R/W PORT — selects the 7712 I/O address (39 hex, as shipped) and the I/O mode (16-, 10-, or 8-bit I/O addressing).

DRAM CONTROLLER — supervises all DRAM access and transparent refresh operations.

RESET — Resets the DRAM controller state machine during power-up and power-down. A SYSRESET* signal clears the MPU and DMA block registers.

REFRESH TIMER — provides timing for transparent refresh.

DYNAMIC MEMORY ARRAY — 64K, 128K, 256K, and 512K byte DRAM configuration options are available. The memory is divided into two banks and a ninth parity bit is provided for parity error detection.

PARITY ERROR CONNECTOR AND LED — user interfaces which indicate a DRAM parity error.

Section 2

INSTALLATION AND SPECIFICATIONS

INTRODUCTION

This section describes the procedure for installing the 7712 card into an STD BUS system. Options and specifications are included that must be considered before operating this card in your system.

Section 3, Operation and Programming, must be read for an explanation of the card's operation before decisions can be made on configuring the jumpers for your application.

INSTALLATION

CAUTION

To prevent possible damage to your STD BUS system, make sure that power is **off** before inserting a card into the rack or before removing a card from the card rack.

The 7712 operates as part of an STD BUS system. It can be installed in any slot of an STD BUS card rack. The card ejector tab must be at the top as illustrated in figure 2-1. It is recommended that the 7712 card be used only in the card rack.

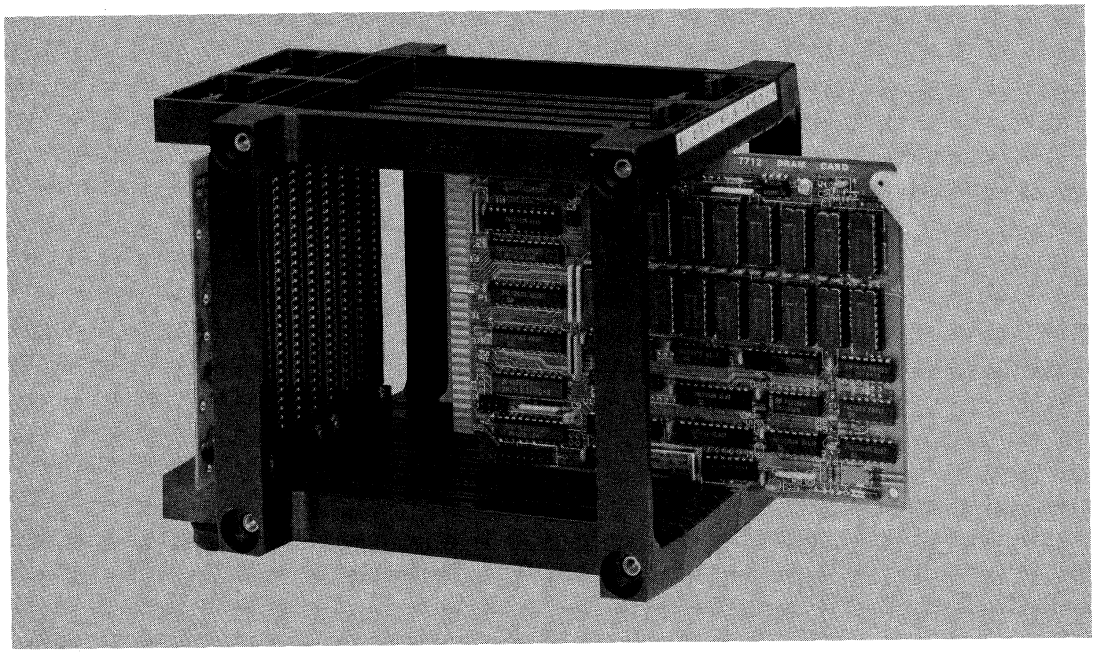


Figure 2-1. Installation (7712)

MEMORY

The 7712 is available fully populated with 512K or 128K bytes of memory or depopulated to 256K or 64K. A combination of the devices may also be used (one bank of 256Kx1 parts and one bank of 64Kx1 parts). To add or replace memory parts, an Intel D2164A-15 (64Kx1), Hitachi HM50256-15 (256Kx1), or equivalent should be used. Additional parts will be qualified by Pro-Log as they become available.

CABLING

There is one external signal associated with the parity checker. This signal is accessed at a two-pin connector on the front of the card which connects the signal to an I/O card. The Pro-Log RC704 cable can be used for this function. The RC704 has a two-pin connector and a twisted pair cable appropriate for carrying one signal and its associated ground.

CARD LAYOUT AND JUMPERS

Figure 2-3, pages 2-4 and 2-5, shows the location of the jumpers, external signal connector, and other features of the 7712. It also gives a brief description of the function of each jumper.

The 7712 decodes the 16 address lines normally associated with 8-bit processors (the STD BUS was designed to support 8-bit processors) or it can decode the 20 address lines normally associated with 16-bit processors (8088). This capability is described further in section 3. The references to 16-bit mode and 8-bit mode in figure 2-3 and elsewhere in this manual refer to these two forms of use.

I/O ADDRESS (W1, W7)

The 7712 offers five I/O port addressing modes:

16-bit (LO) for the 8088 (W1 1-2)
16-bit (HI) for the 8088 (W1 3-4)
10-bit for IBM PC compatibility: 8088 (W1 7-8)
8-bit (IOEXP = X) for the 8085 (W1 No Jumper)
8-bit (IOEXP = 0) for the 8085 (W1 5-6)

One (only) of these modes must be selected using jumper W1. See section 3, I/O Port Addressing, for more information.

The 7712 occupies one I/O port address (39 hex) as shipped. An I/O Read at 39 hex reads the status of the parity from data line 0 (i.e., if D0 = 1, no parity error; if D0 = 0, then parity error). An I/O Write at 39 hex clears any parity error. An I/O Write at the selected address also affects the block registers, depending on which of the three block modes the user selects (see section 3, Block Register). The I/O address can be changed to any one of 256 I/O addresses (00, 01, 02 ... FE, FF) by jumper group W7 (see figures 2-2 and 2-3).

All 7712 cards in a system must use the same I/O addressing mode (W1). Whenever the parity error status is read (i.e., Parity Error generates INTRQ* W2 1-2), the I/O address must be different for each card (W7); otherwise, all I/O addresses can be the same.

PINS	1-2	3-4	5-6	7-8	9-10	11-12	13-14	15-16	
BITS	D7	D6	D5	D4	D3	D2	D1	D0	
JUMPERS	0	0	1	1	1	0	0	1	39 hex

0 = Jumper in
1 = Jumper out

Figure 2-2. W7 I/O Address Jumper (7712)

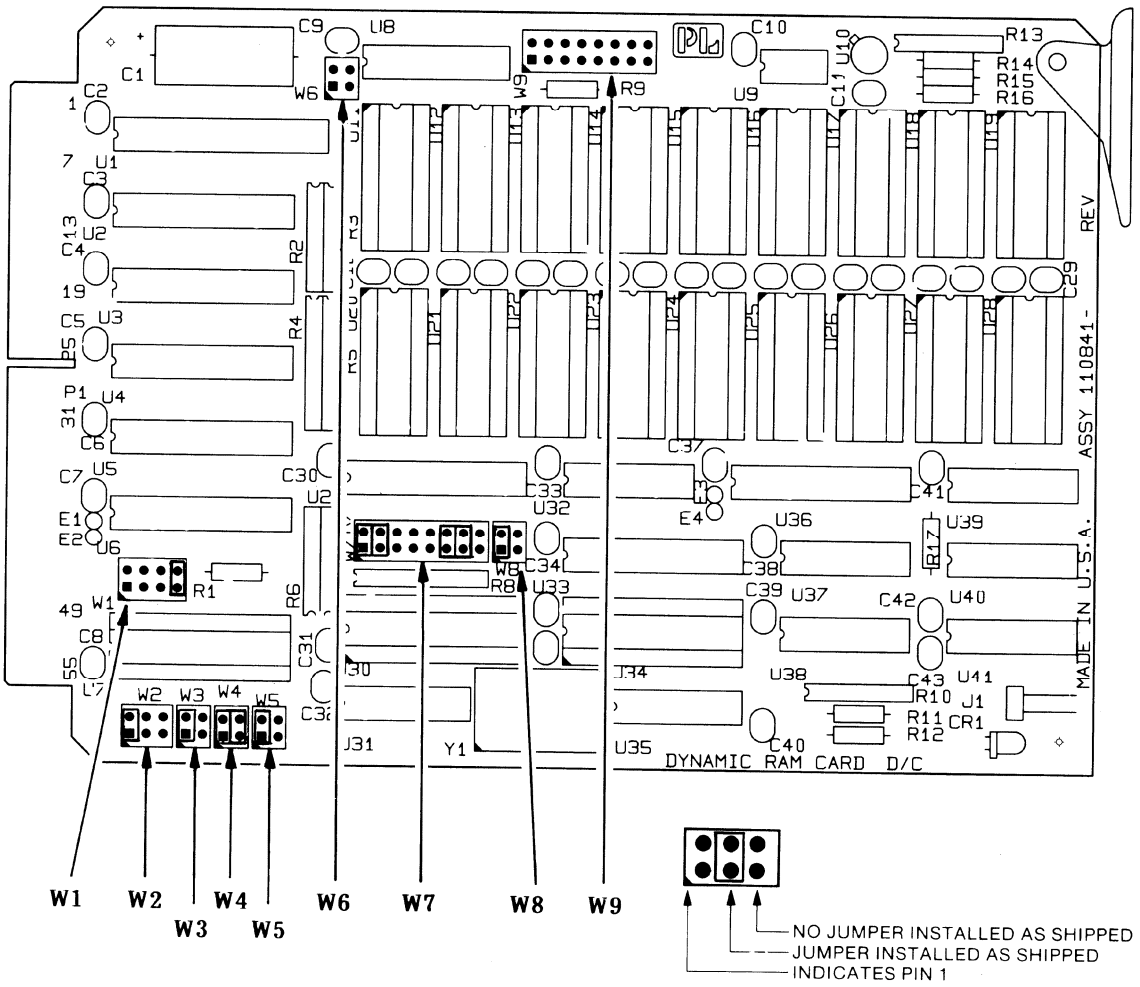


Figure 2-3. Card Layout and Jumper Descriptions (7712)

JUMPER	PINS	DESCRIPTION	Reference Page Numbers
W1		I/O MODE JUMPER selects:	2-2, 3-1, 3-2, 3-4
	1-2	16-bit Lo addressing (8088)	
	3-4	16-bit Hi addressing (8088)	
	5-6	8-bit addressing with IOEXP=0 (8085)	
	No jumper	8-bit addressing with IOEXP= don't care (8085)	
	7-8*	10-bit addressing for IBM PC compatibility (8088)	
W2		PARITY ERROR OUTPUT JUMPER selects:	2-7, 3-15 to 3-17
	1-2*	INTRQ* in response to parity error	
	3-4	NMIRQ* in response to parity error	
	5-6	PBRESET* in response to parity error (debounced PBRESET* circuits may use 1-2 also)	
	No jumper	No response to parity error	
W3		BLOCK REGISTER ENABLE JUMPER selects:	2-7, 3-4 to 3-14
	1-2*	DMA block register when BUSAK* is active. (For use with 16-bit addressing DMA processors in a 20-bit 8088 system)	
	No jumper	DMA register disable (when using 20-bit DMA cards)	
	3-4	8085 64K block mode	
W4		BLOCK SIZE JUMPER selects:	2-7, 3-4 to 3-14
	1-2,3-4*	64K block size (8088 or 8085)	
	1-3,2-4	32K block size (8085 only)	
W5		W5 has a different function depending on the block mode selected (see figure 2-4)	2-6, 2-7, 3-4 to 3-14
		DRAM ENABLE JUMPER enables 7712 in:	
	1-2*	Lower half of 8088 address space (00000-7FFFF) or Lower half of 8085 address space (0000-7FFF), 32K block size only or Do not use in 8085 64K block size mode	
	3-4	Upper half of 8088 address space (80000-FFFFF) or Upper half of 8085 address space (8000-FFFF), 32K block size only or 8085 64K block mode	
W6		DEVICE SIZE JUMPER selects:	2-7
	1-2	64K x 1 dynamic RAM parts	
	3-4	256K X 1 dynamic RAM parts	
W7	(39H)*	I/O ADDRESS JUMPER selects 1 of 256 locations where pins 1-2 are MSD and 15-16 are LSD	2-2, 2-3
W8		MPU JUMPER selects:	2-7
	1-2*	8088	
	3-4	8085	
W9		BLOCK (SEGMENT) JUMPER enables blocks on 64K boundaries. Pins 1-2 are LSD and 15-16 are MSD. (See figure 2-4.)	2-3, 2-6 3-4 to 3-14

*As shipped configuration

Figure 2-3. (Continued)

MEMORY ADDRESS (W5, W9)

The memory address range occupied by the 7712 is controlled by jumper group W5 and W9. The configuration of these jumpers is different for 8-bit (8085) and 16-bit (8088) systems. First select a block mode for all cards (see section 3, Memory Management, page 3-2) and then establish the address range for each of the cards (see figure 2-4).

JUMPERS		BLOCK (SEGMENT) MODES						7712 CARDS			
		8088 (64K Blocks)		8085 (32K Blocks)		8085 (64K Blocks)		(Versions may be mixed)[2]			
W5	W9	SEG	ADDRESS	BLK[1]	ADDRESS	BLK	ADDRESS	-512K	-256K	-128K	-64K[3]
1-2	1-2	0	00000-0FFFF	0	DRAM MAPS INTO MPU ADDRESS SPACE 0000-7FFF	NOT APPLICABLE		#1	#1	#1	#1
	3-4	1	10000-1FFFF	1							#2
	5-6	2	20000-2FFFF	2						#2	
	7-8	3	30000-3FFFF	3							
	9-10	4	40000-4FFFF	4					#2	#3	
	11-12	5	50000-5FFFF	5							#3
	13-14	6	60000-6FFFF	6						#4	
	15-16	7	70000-7FFFF	7							
3-4	1-2	8	80000-8FFFF	0	DRAM MAPS INTO MPU ADDRESS SPACE 8000-FFFF	0 1 2 3 4 5 6 7 BLOCK REG. DATA BITS[4]		#2	#3	#5	
	3-4	9	90000-9FFFF	1							
	5-6	A	A0000-AFFFF	2						#6	
	7-8	B	B0000-BFFFF	3							
	9-10	C	C0000-CFFFF	4					#4	#7	#4
	11-12	D	D0000-DFFFF	5							
	13-14	E	E0000-EFFFF	6						#8	
	15-16	F	F0000-FFFFF	7							

[1] These are 64K blocks. Data bit 3 of Block Register selects LO or HI half of 64K block (D3 = 0 — 0000-7FFF, D3 =1 — 8000-FFFF).

[2] 8088 maximum system memory = 1 megabyte, 8085 maximum DRAM memory = 512K bytes with either W5 = 1-2 or 3-4.

[3] 64K versions are not recommended for large amounts of memory. A random block mapping example is shown for the 8088.

[4] D2-D0 selects upper 32K block; D3 enables upper block. D6-D4 selects lower 32K block; D7 enables lower block.

Figure 2-4. Memory Address Mapping (7712)

BLOCK MODE (W3, W4, W5)

The 7712 has three block modes: 64K blocks for the 8088, and 64K and 32K blocks for the 8085. Refer to section 3, Memory Management, page 3-2, for a discussion of memory mapping options. Jumpers W3, W4, and W5 control these modes and are functionally interrelated (see figure 2-3, page 2-5).

MEMORY PARTS (W6)

Jumper W6 selects the memory device size of each bank of memory. W6 1-2 (as shipped) selects 64Kx1 dynamic RAM parts and W6 3-4 selects 256Kx1 parts. If each memory bank has different components (one bank with 64K parts and one bank with 256K parts), jumper W6 should be in the 3-4 position. Memory parts cannot be mixed within the banks.

PROCESSOR TYPE (W8)

Jumper W8 selects the processor type (W8, 1-2 for the 8088; W8, 3-4 for the 8085).

PARITY ERROR DETECTION (W2)

The memory array on the 7712 is nine bits wide. Whenever a byte is written to the 7712, a parity bit is generated and stored in the ninth bit. Whenever a byte is read from a 7712, this parity bit is checked. If an error has occurred, the 7712 can so indicate.

Jumper W2 (figure 2-3, page 2-4) determines the 7712 response. There are five responses possible: nonmaskable interrupt (W2 3-4), maskable interrupt (W2 1-2), reset (W2 5-6), reset and interrupt (W2 1-2 and 5-6), or no response (no jumper). All of these responses in addition to the parity error LED, are discussed in section 3.

SPECIFICATIONS

Mechanical Specifications

The 7712 meets all general mechanical specifications of the STD BUS.

Figures 2-5 through 2-11 on the following pages give specifications for the 7712.

PIN NUMBER / MNEMONIC	OUTPUT (LSTTL DRIVE)	INPUT (LSTTL LOADS)	PIN NUMBER / MNEMONIC	OUTPUT (LSTTL DRIVE)	INPUT (LSTTL LOADS)
PE*		15	2	1	GROUND

Figure 2-5. Connector J1 (7712)

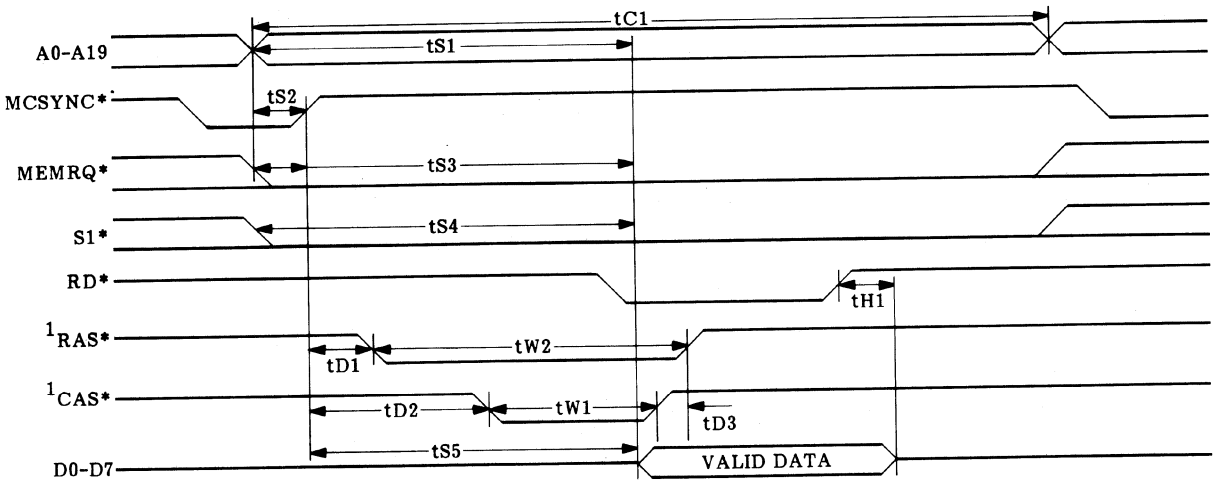
Electrical Specifications

	RECOMMENDED OPERATING LIMITS			ABSOLUTE NONOPERATING LIMITS		
PARAMETER	**MIN**	**TYP**	**MAX**	**MIN**	**MAX**	**UNITS**
Supply Voltage	4.75	5.00	5.25	0	7.00	Volts
Supply Current 128K	—	920	1480	—	—	mA
Supply Current 512K		950	1500	—	—	mA

Environmental Specifications

	RECOMMENDED OPERATING LIMITS			ABSOLUTE NONOPERATING LIMITS		
PARAMETER	**MIN**	**TYP**	**MAX**	**MIN**	**MAX**	**UNITS**
Free Air Temperature	0	+25	+65	-40	+75	°C
Noncondensing Humidity	5	—	95	5	95	%RH

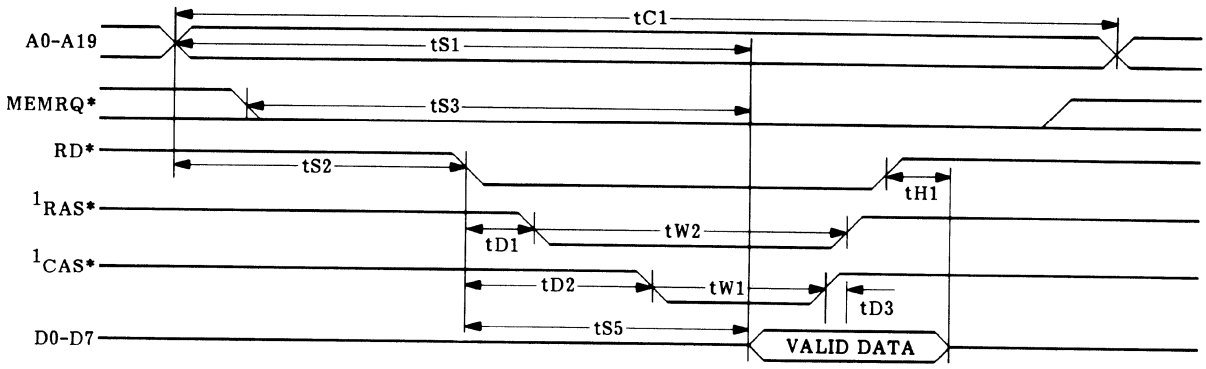
Figure 2-6. Electrical and Environmental Specifications (7712)



SYMBOL	PARAMETER	MIN	MAX
tC1	MPU read cycle time	360	
[1]tD1	RAS* delay from MCSYNC* inactive		37
[1]tD2	CAS* delay from MCSYNC* inactive		182
[1]tD3	CAS* to RAS* inactive		50
tH1	Data hold time after RD* inactive	3	
tS1	Address setup before valid data	376	
tS2	Address setup before MCSYNC* inactive	15	
tS3	MEMRQ* setup before valid data	316	
tS4	S1* setup before valid data	302	
tS5	MCSYNC* setup before valid data	288	
[1]tW1	CAS* pulse width	150	
[1]tW2	RAS* pulse width	200	

All times given in nanoseconds.
[1] All times given at the STD BUS except where indicated.

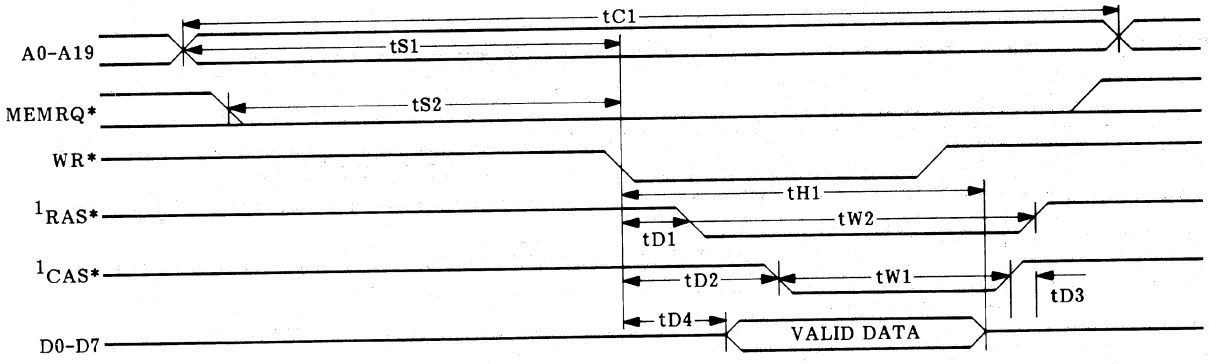
Figure 2-7. Processor Read Timing (7712)



SYMBOL	PARAMETER	MIN	MAX
tC1	DMA read cycle time	450	
[1]tD1	RAS* delay from RD*		51
[1]tD2	CAS* delay from RD*		221
[1]tD3	CAS* to RAS* inactive		50
tH1	Data hold time after RD* inactive	3	
tS1	Address setup before valid data	401	
tS2	Address setup before RD*	93	
tS3	MEMRQ* setup before valid data	341	
tS5	RD* setup before valid data	327	
[1]tW1	CAS* pulse width	150	
[1]tW2	RAS* pulse width	198	

All times given in nanoseconds.
[1] All times given at the STD BUS except where indicated.

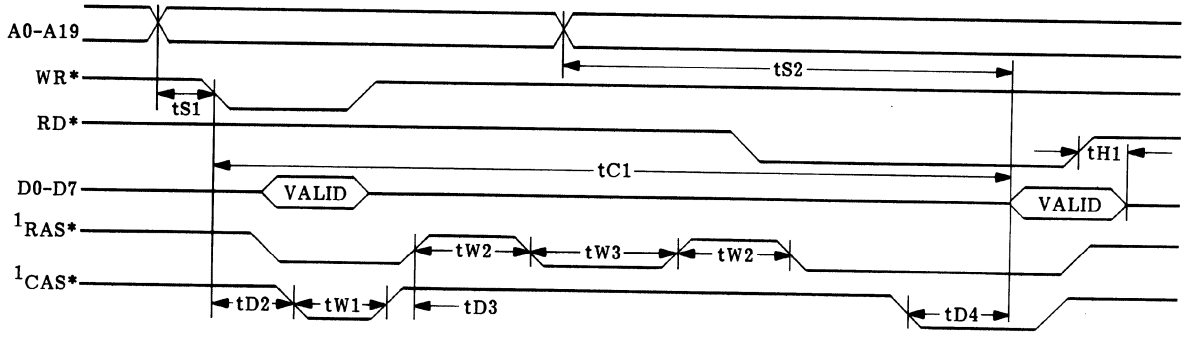
Figure 2-8. DMA Read Timing (7712)



SYMBOL	PARAMETER	MIN	MAX
tC1	Write cycle time	450	
1tD1	RAS* delay from WR*		51
1tD2	CAS* delay from WR*		221
1tD3	CAS* to RAS* inactive		50
tD4	Valid data delay from WR*		30
tH1	Data hold time after WR* (MPU)	235	
tH1	Data hold time after WR* (DMA)	260	
tS1	Address setup before WR*	91	
tS2	MEMRQ* setup before WR*	31	
1tW1	CAS* pulse width	100	
1tW2	RAS* pulse width	157	

All times given in nanoseconds.
1 All times given at STD BUS except where indicated.

Figure 2-9. Write Timing (7712)



SYMBOL	PARAMETER	MIN	MAX
tC1	Forced Refresh cycle time	847	
1tD1	RAS* delay from WR*		51
1tD2	CAS* delay from WR*		196
1tD3	CAS* to RAS* inactive		25
tD4	Valid data delay from CAS* (t_{CAC})		85
tH1	Data hold time after RD* inactive	3	
tS1	Address setup time before WR*	91	
tS2	Address setup time before valid data	376	
1tW1	CAS* pulse width	100	
1tW2	RAS* precharge pulse width	100	
1tW3	RAS* pulse width during Refresh	150	

All times given in nanoseconds.
1 All times given at STD BUS except where indicated.

Figure 2-10. Forced Refresh Timing (7712)

PIN NUMBER	OUTPUT (LSTTL DRIVE)	INPUT (LSTTL LOADS)	MNEMONIC	PIN NUMBER	OUTPUT (LSTTL DRIVE)	INPUT (LSTTL LOADS)	MNEMONIC
+5V	IN		2	1		IN	+5V
GROUND	IN		4	3		IN	GROUND
-5V			6	5			-5V
D7	2	55	8	7	55	2	D3
D6	2	55	10	9	55	2	D2
D5	2	55	12	11	55	2	D1
D4	2	55	14	13	55	2	D0
A15	1		16	15		2	A7
A14	1		18	17		2	A6
A13	1		20	19		2	A5
A12	1		22	21		2	A4
A11	1		24	23		2	A3
A10	1		26	25		2	A2
A9	2		28	27		2	A1
A8	2		30	29		2	A0
RD*	1		32	31		1	WR*
MEMRQ*	1		34	33		1	IORQ*
MEMEX			36	35		1	IOEXP
MCSYNC*	2		38	37			REFRESH*
STATUS 0*			40	39		2	STATUS 1*
BUSRQ*			42	41		1	BUSAK*
INTRQ*		50	44	43			INTAK*
NMIRQ*		50	46	45	50		WAITRQ*
PBRESET*		50	48	47		1	SYSRESET*
CNTRL*			50	49			CLOCK*
PCI	IN		52	51	OUT		PCO
AUX GND			54	53			AUX GND
AUX -V			56	55			AUX +V

* Low-level active logic

Figure 2-11. STD Edge Connector Pin List (7712)

INTRODUCTION

This section discusses the 7712 operating modes. The text and graphics should aid the user in understanding the different functional modes of the 7712 I/O port addressing, 8088 and 8085 memory management, block register operation in each mode, parity error detection, refresh modes, and DMA controller compatibility.

I/O PORT ADDRESSING

The 7712 card provides five modes of I/O port addressing. The mode selected depends on the I/O port addressing capability required in the application. The 8088 processor uses 16 lines to decode port addresses and the 8085 uses 8-bit addressing. Although the IBM PC contains an 8088 processor, it uses only ten bits to address ports. The 7712 can be mapped at one of only 256 ports. For 8-bit processors (with 8-bit I/O addressing), each I/O port address selects a unique port. In 10- and 16-bit address systems, however, each 7712 card port address redundantly appears through portions of the I/O map.

1. 8-bit I/O port addressing with IOEXP (8085)

   With W1 5-6 in place and IOEXP low, the 7712 port is enabled at its selected I/O address (W7). With IOEXP high, the 7712 port is disabled.

2. 8-bit I/O port addressing with IOEXP "don't care"

   If IOEXP is unused or is a "don't care," the W1 jumper should not be used. A pull-down resistor guarantees that IOEN* is held low. The 7712 port is enabled at its selected I/O address (W7) regardless of the state of IOEXP.

3. Partial 16-bit I/O (LO) addressing (8088)

   With the W1 jumper at 1-2, the 7712 port address is redundantly mapped every 256 addresses in the lower half of the 64K port addresses. The 7712 port is not mapped in the upper 32K (see figure 3-1).

4. Partial 16-bit I/O (HI) addressing

   With the W1 jumper at 3-4, the 7712 is redundantly mapped in the upper half of the 64K port addresses and is not mapped in the lower 32K (see figure 3-2).

5. 10-bit I/O port addressing (IBM PC)

   With the W1 jumper at 7-8, the 7712 port address is redundantly mapped in one of the lower 256 addresses of each 1K address block (see figure 3-3). The port is enabled only if both address A8 and A9 are low.

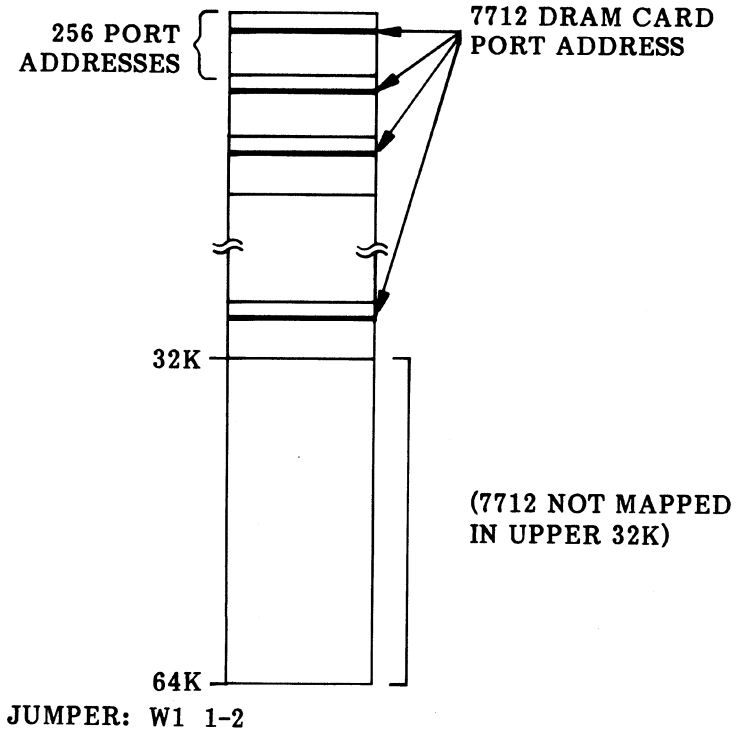


Figure 3-1. 16-Bit I/O Port Addressing (Lower 32K)

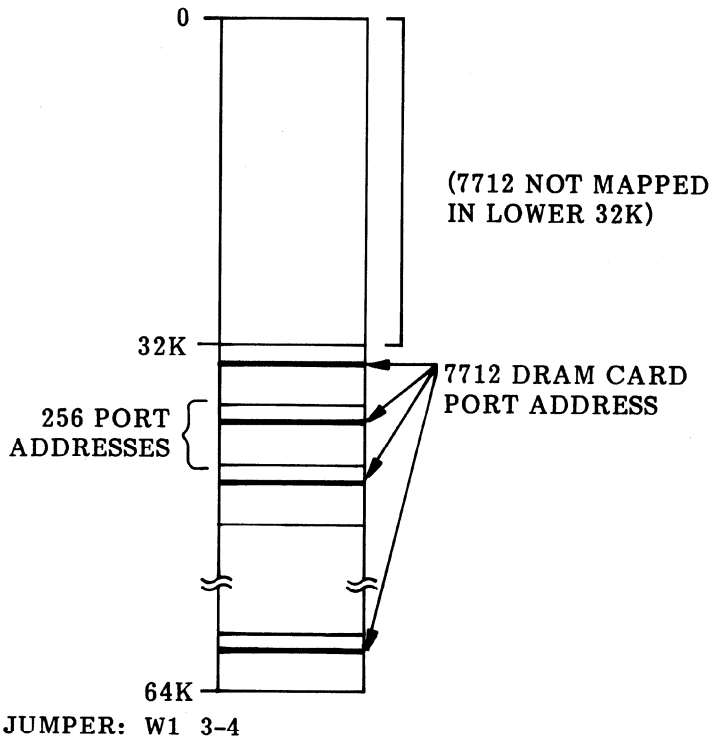


Figure 3-2. 16-Bit I/O Port Addressing (Upper 32K)

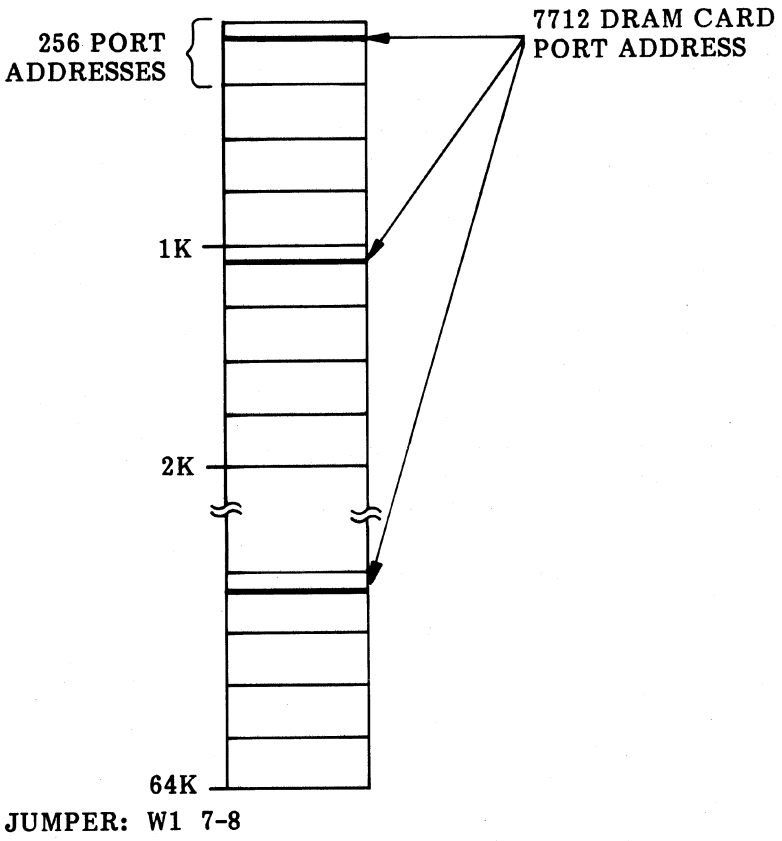


Figure 3-3. 10-Bit I/O Port Addressing

MEMORY MANAGEMENT

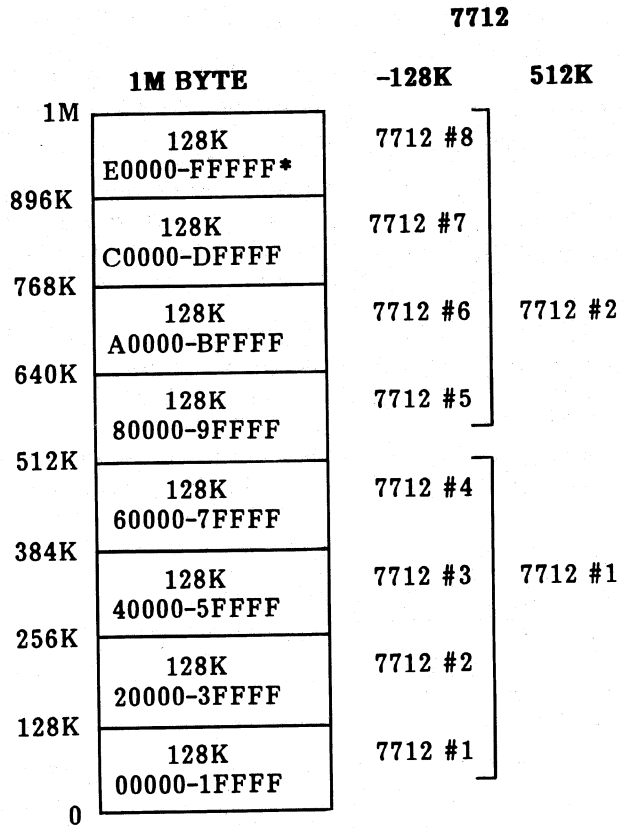
The 7712 operates with an 8085 processor on the STD BUS or an 8088 processor multiplexed onto the STD BUS. Memory management on the 7712 is significantly different for these two types of systems. Read only the section appropriate for your application. To gain more technical knowledge of the 7712, you may wish to read both sections. Before doing this, reach a thorough understanding of the mode appropriate for your application.

16-Bit Systems — 8088

The 7712 is shipped prepared for use with the 8088. Each 7712 card needs a different set of segment (block) addresses. The 8088 uses 20-bit addressing to directly access up to one megabyte consisting of two 7712-512K, four 7712-256K, eight 7712-128K, or sixteen 7712-64K cards — or some combination of these. Naturally it is best to use the least amount of cards possible. The 7712 provides 64K data segments (memory blocks) which are individually enabled with jumper group W9. The 7712 must be be enabled in either the lower 512K (00000-7FFFF as shipped) or upper 512K (80000-FFFFF) of memory with jumper W5 3-4. In systems with multiple 7712 cards, the user must ensure each card is mapped to a different area to prevent memory contention. Figure 2-4, page 2-6, shows the jumper placements for 8088 memory mapping. **NOTE:** The 8088 card should be configured to use normal write mode, not advanced.

Special precautions are required when using memory cards without 20-bit addressing capability. These cards will be redundantly mapped throughout the entire one megabyte address space. If these cards respond to MEMEX, and A19 controls MEMEX (Pro-Log 8088 CPU cards), then a limited system can be implemented. All 7712 cards must be enabled in the lower megabyte (A19 = MEMEX = 0) and all non-20-bit memory cards in the upper megabyte (A19 = MEMEX = 1) or vice versa.

Each of the sixteen 64K segments in the one megabyte address range can be enabled or disabled with jumpers W5 and W9. This allows flexibility in locating other cards in the one megabyte memory space. The memory on other cards must be at least part ROM, especially at the processor reset address. In many systems, the only memory not on the 7712 will be a bootstrap ROM in the reset address range. Figure 3-4 shows the memory map for 20-bit addressing systems. The DRAM memory appears to the 8088 as one continuous block of memory. No special memory management need be performed.



* **NOTE:** The 7712 may be enabled at address F0000-FFFFF if the processor partially masks the DRAM off the 8088 internal Data bus. Pro-Log's 8088 MPU cards (7861 and 7862) provide this feature.

Figure 3-4. Memory in 16-Bit Systems (7712)

8088 Segment Mode Select: All 512K bytes of memory on the 7712 card are directly accessed in a 20-bit system by demultiplexing the upper addresses from the data lines. The 8-bit block register is used to select the active portion of memory. The register is divided into two 4-bit sections each with different purposes and each with opposite output enables.

For systems with 20-bit addressing, the lower half of the block register (D0-D3) is the MPU register (see figure 3-5). It demultiplexes address lines A16 through A19 from the data bus. This lower half is enabled for normal operation and for 20-bit DMA controller applications.

The upper half (D4-D7) is used for 16-bit address DMA controllers in an 8088 system. An I/O write to this port (address 39 hex as shipped) selects one of 16 memory segments. This is the segment that is accessed during 16-bit DMA.

In all modes, an I/O write to the port clears the parity status, and an I/O read from the port provides the parity status on data line 0 (if D0 = 1, then no parity error; if D0=0, parity error exists).

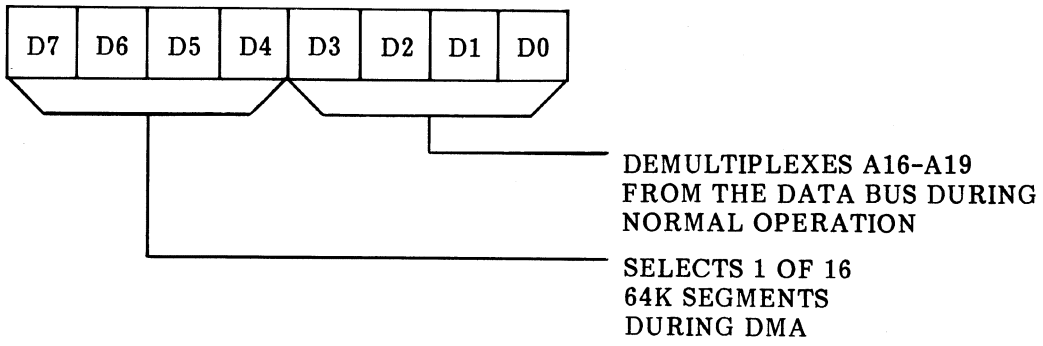


Figure 3-5. 8088 Segment Mode Register

8-Bit Systems — 8085

A 7712 card may have up to 512K bytes of memory or up to four 128K cards in a system (512K total). There may also be up to 64K of memory on other cards for a system total of 576K. An 8-bit processor can access only 64K at a time, so memory must be divided into blocks and made available to the processor one block at a time. In a 16-bit address system, both a block of memory and the address in the block must be selected. This allows 1/2 Megabyte of memory to be accessed in a system that can directly address only 64K bytes. On the 7712, the memory is divided into 32K or 64K-byte blocks and is selected in the block register by the onboard I/O port. Refer to figures 2-3 and 2-4, pages 2-4 and 2-6, for jumper configurations of the 7712 in 8085 systems.

8085 32K Block Mode Select: Memory divided into 32K blocks allows 16 blocks on the 7712 and two blocks on other cards (see figure 3-7). The blocks can be selected in almost any combination and made available to the processor one block at a time.

For normal operation, bits D0-D2 of the block register select one of the eight 64K memory blocks to appear in the MPU memory address space (see figure 3-8). D3 selects the upper 32K (D3 = 1) or the lower 32K (D3 = 0) of the 64K block. Address lines A0-A14 select the memory location accessed within the 32K block. Address line A15 selects the MPU memory (A15 = 0) or the DRAM memory (A15 = 1). For 16-bit DMA applications, D4-D7 selects which of the sixteen 32K memory blocks is accessed. D7 selects the upper 32K (D7 = 1) or the lower 32K (D7 = 0) of the 64K block.

An I/O write to the port (39 hex as shipped) configures the DMA and MPU register. In all modes, an I/O write to the port clears the parity status, and an I/O read from the port provides parity status on bit D0 (D1-D7 are "don't cares").

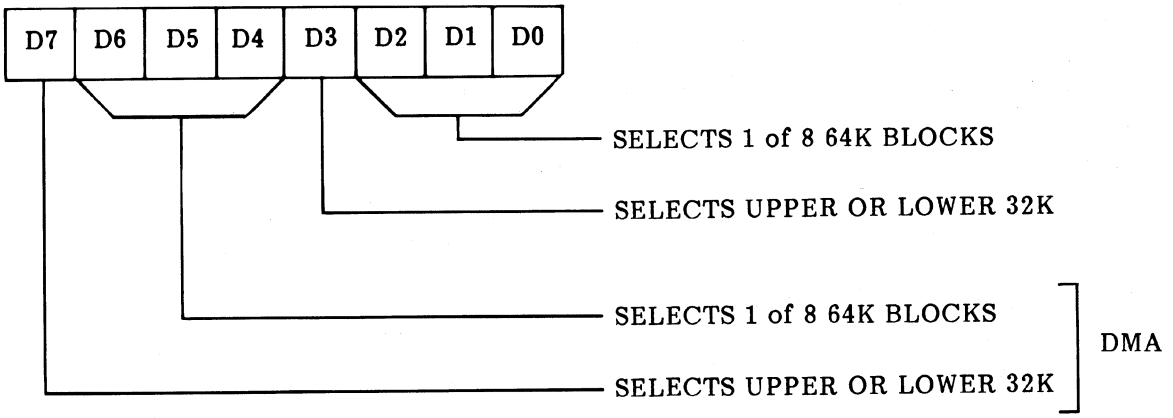


Figure 3-6. 8085 32K Block Mode Register

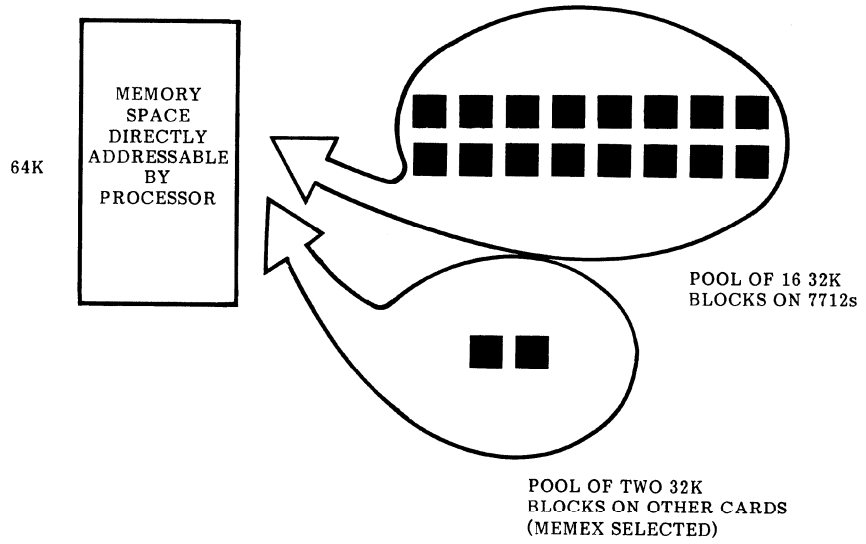

64K
MEMORY SPACE DIRECTLY ADDRESSABLE BY PROCESSOR
POOL OF 16 32K BLOCKS ON 7712s
POOL OF TWO 32K BLOCKS ON OTHER CARDS (MEMEX SELECTED)

Figure 3-7. Pool of Memory Blocks (7712)

Figures 3-8 and 3-9 show the two most common ways to draw on the memory pool. In both Basic Arrangements 1 and 2, jumpers W4 1-3 and 2-4, W8 3-4, and W9 are connected.

In Basic Arrangement 1, one or two blocks of memory on other cards in the system occupy the lower addresses (controlled by MEMEX). This memory could be all ROM or a combination of ROM and RAM. The memory on the 7712 is used only in the upper addresses. This kind of arrangement might be used in a dedicated control situation in which a ROM-based program requires a large amount of RAM space for data collection. The program would reside in ROM in the lower addresses while the 7712 would provide 16 blocks of RAM in the upper addresses that the processor could access as needed. Jumper W5 3-4 is connected for this configuration.

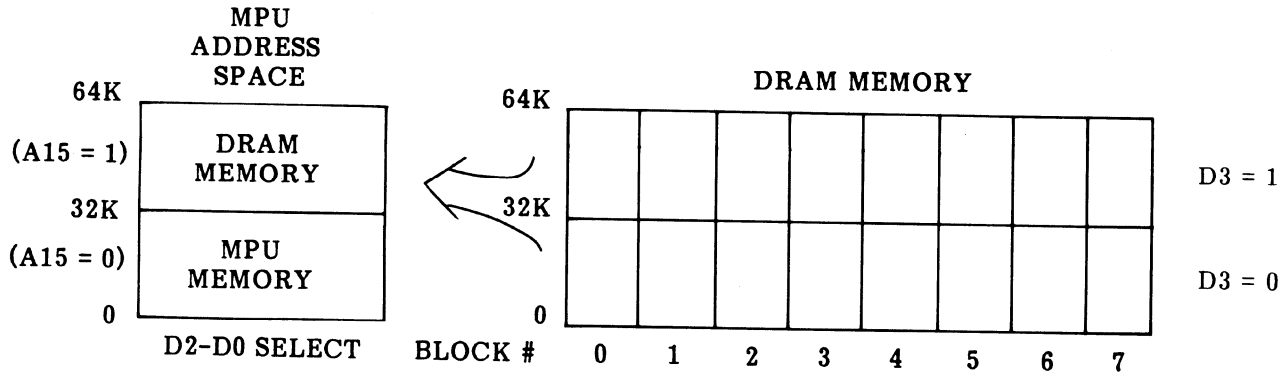


Figure 3-8. Basic Arrangement 1 (7712)

Basic Arrangement 2 is similar to Arrangement 1 except the DRAM blocks are mapped to the lower 32K of the MPU address space. Arrangement 2 is useful in applications where the MPU deselects offboard memory after reset. W5 1-2 is connected for this configuration.

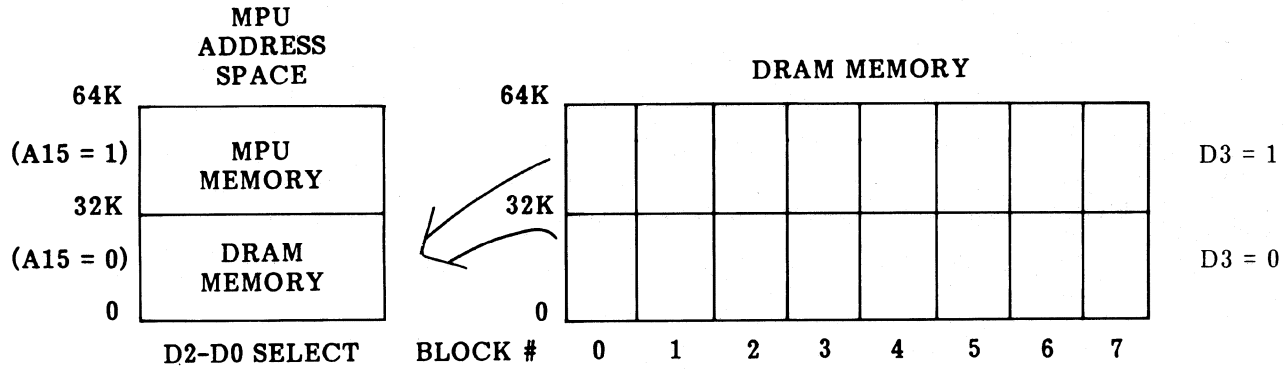


Figure 3-9. Basic Arrangement 2 (7712)

In figure 3-10a, Basic Arrangement 1 shows the conceptual structure of the memory. Basic Arrangement 2 reverses the MPU and DRAM enable space, but otherwise is identical to this discussion and the illustrations in figure 3-10.

The memory on the 7712 is used only in the upper addresses. The two blocks of memory on other cards in the system occupy only the lower addresses and must be selected with MEMEX HI and LO. MEMEX is the memory expansion line of the STD BUS. In a system, it allows two banks of memory to exist at the same address, only one of which is enabled at a time depending on the state of the MEMEX line. Figures 3-10b and 3-10c show the ways MEMEX can be used in conjunction with Basic Arrangements 1 and 2. Control of the 7712 memory is performed solely by means of the onboard I/O port. The 7712 does not decode MEMEX. To implement some of the following examples, memory on the other cards must decode MEMEX.

No MEMEX: In figure 3-10b, the memory on other cards is limited to one block of 32K and does not respond to MEMEX at all; thus it is permanently enabled. Any of the sixteen 7712 blocks can be selected to be used with the other memory. To use the memory in this form, configure the memory on other cards to disregard MEMEX. The cards which can drive MEMEX should be checked to ensure only one of them is driving MEMEX. More than one card driving MEMEX results in bus

contention; no card driving MEMEX leaves the line floating, which is bad practice. The lowest addresses of the permanent memory need to be ROM for bootstrap memory. For systems that start at the top of memory, the permanent memory is in the upper 32K.

Two-Block MEMEX: The Two-Block MEMEX system (figure 3-10c) is similar to the No MEMEX system except there are two 32K blocks of memory on other cards. Both of them reside in the lower addresses and only one is enabled at a time, depending upon the state of MEMEX. Most Pro-Log processor cards have an onboard I/O port for controlling MEMEX. Only one card in the system should drive MEMEX. Some bootstrap ROM has to reside in at least the lowest addresses of the bank of MEMEX-controlled memory that is enabled at power-up.

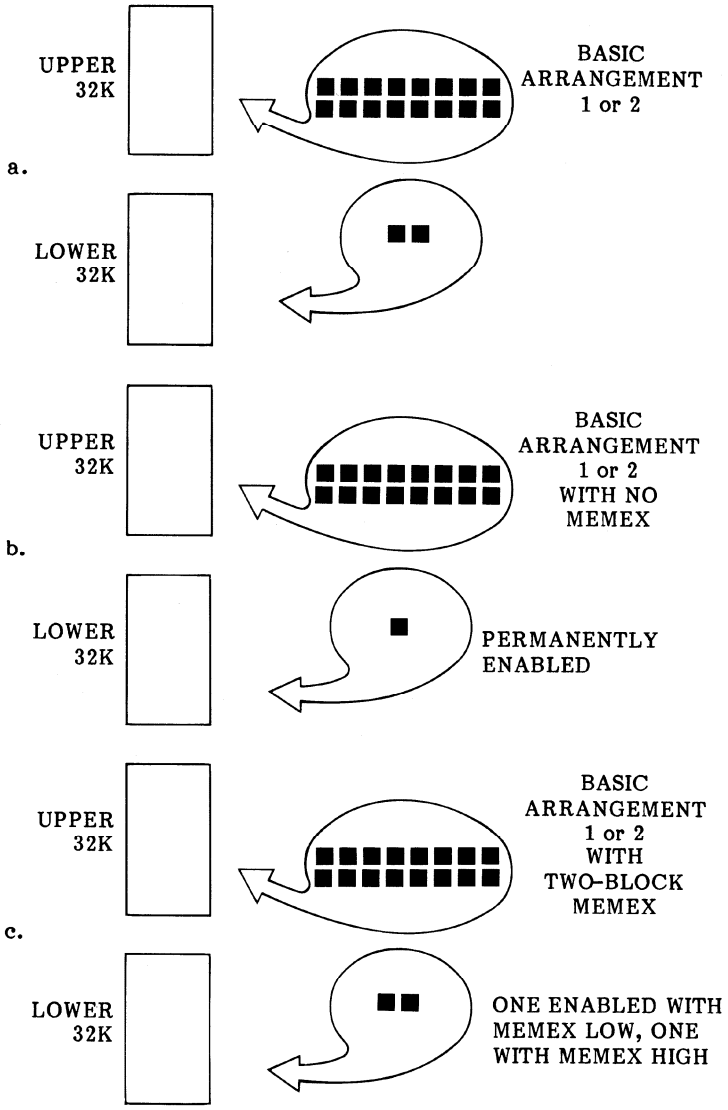


Figure 3-10. Basic Arrangements 1 and 2 (7712)

8085 64K Block Mode Select: Memory management in the 64K mode is configured by an I/O write to the port (address 39 hex as shipped). Bit D3 enables the eight upper 32K blocks, and bits D0-D2 select one of the eight. Bit D7 enables the eight lower 32K blocks, and bits D4-D6 select one of the eight (see figure 3-11). As in the other modes, an I/O write to the port clears the parity status and an I/O read from the port provides parity status on D0.

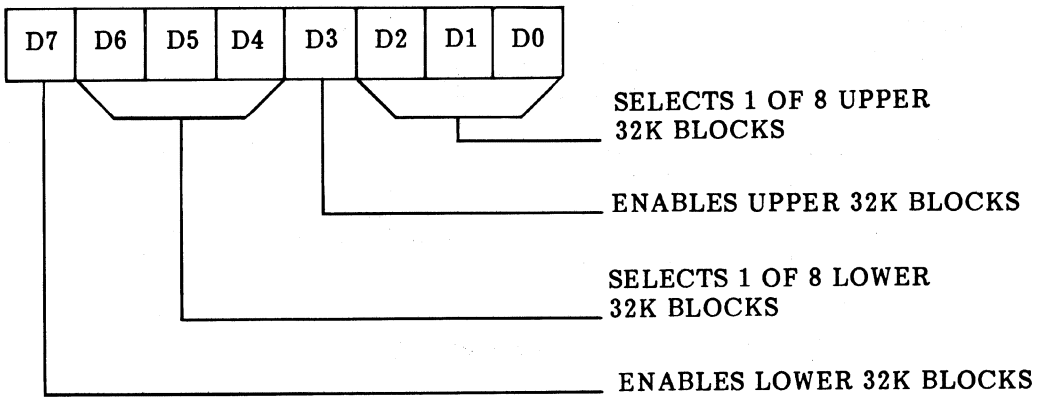


Figure 3-11. 8085 64K Block Mode Register

Figure 3-12 shows the third basic form of use for the 7712 memory. Basic Arrangement 3 is similar to the One-Block MEMEX of figure 3-10B. Memory can be used as eight blocks of 64K instead of sixteen blocks of 32K. The 32K blocks are still independent and may be arranged in any combination. The jumper configurations for this mode are W3 3-4, W4 1-2 and 3-4, W5 3-4, W8 3-4, and W9.

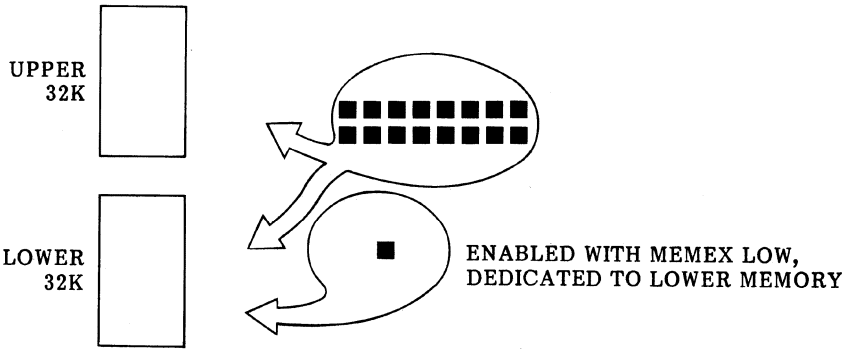


Figure 3-12. Basic Arrangement 3 (7712)

Figure 3-13 shows the memory used as if it were all in 64K blocks. This configuration is used when a system, once booted, operates solely in DRAM.

In figures 3-13 and 3-14, DUy represents the upper 32K DRAM of block number y (y = 0-7). DLx is the lower 32K of block number x (x = 0-7). If the DUy enable bit is set (D3 = 1) and the lower register (DLx) is inactive (D7 = 0) when address A15 is high, the upper register (DUy) is active. If the DLx enable bit is set (D7 = 1) and DUy is inactive (D3 = 0) when A15 = 0, DLx is active. If both bits (D3 = D7 = 1) are set, the DLx is active when A15 = 0 and DUy is active when A15 = 1. Thus, a full 64K bytes of DRAM are active in an 8085 system.

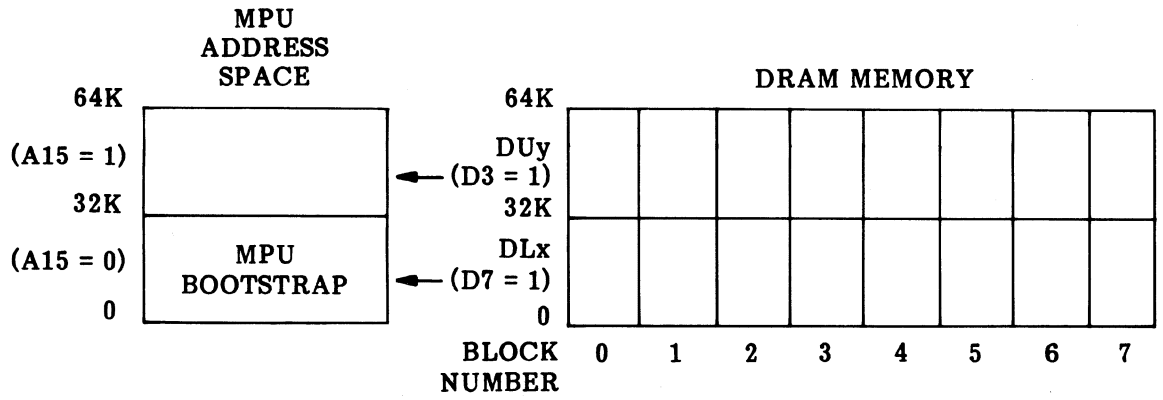


Figure 3-13. 8085 64K Block Mode (7712)

This arrangement might be used in a small general purpose disk-based computer. The memory in other cards may consist of a small amount of bootstrap ROM. Once running, the system operates out of the 64K blocks of RAM on the 7712, dedicating each of sixteen 32K-blocks to a different user and switching between the blocks as it timeshares the system. The memory still consists of independent 32K blocks, but is used in independent pairs as if it were in 64K blocks.

Three software steps are required to enable the 7712 in the 64K mode. At reset, no DRAM is enabled. The 8085 bootstrap PROM is in control. This ROM must be configured to be enabled when the STD BUS MEMEX line is low. All Pro-Log processor cards can control MEMEX. Figure 3-14 demonstrates the method oftransferring control from the bootstrap PROM to DRAM.

1. The bootstrap program must write to the 7712 I/O port (39 hex, as shipped) to enable one of the eight possible 32K DRAM upper blocks (DUy). The Disk Operating System (DOS) can be loaded into this area and control passed to the DOS program.

2. The DOS program in DUy must write to the MEMEX control port to disable the boostrap PROM.

3. The DOS program can now enable any lower DRAM block DLx by writing to the 7712 I/O port with new data.

After power-up or reset, the bootstrap memory is enabled (see figure 3-14a). Write a 08 hex to the Block Select port, port address 39 hex as shipped (figure 3-14b). This enables half of the 7712 memory in the upper half of the processor 64K address range (Block 0). The bootstrap program in ROM loads the control program into the RAM in the upper half of memory. When the processor jumps to the upper half of memory and runs the control program, it writes to the MEMEX control port to disable the bootstrap PROM (figure 3-14c). Then the program writes an 88 hex to the processor memory range (Block 0). The upper and lower block numbers can be changed with additional 7712 I/O writes.

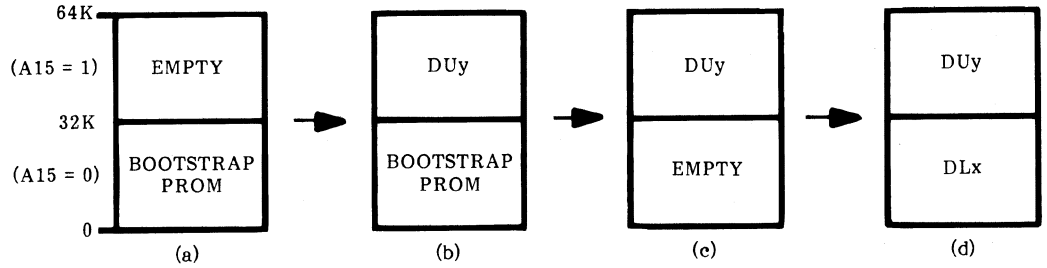


Figure 3-14. 8085 64K Block Enable (7712)

The arrangements shown and their variations are conceptual ideas. The sixteen 7712 blocks can be arranged in any combination through software.

PARITY ERROR DETECTION

Introduction

Parity error detection is implemented in dynamic memory systems due to DRAM dependency on refresh. If the refresh circuit fails or is interrupted for any reason, the DRAM will not hold the data. Dynamic RAM is susceptible to high levels of alpha particle radiation. An alpha hit is rare and will cause only a soft DRAM error, which is recoverable. The 7712 card contains parity error detection circuitry in addition to the most advanced refresh generator design.

The 7712 memory array is nine bits wide. Whenever the 7712 is written to, its hardware generates a parity bit which is stored in the ninth bit. Whenever a byte is read, the parity is again generated and compared to the stored parity bit. A mismatch will occur if the stored byte or the parity bit are in error. Hardware on the card interrupts or resets the processor and lights an LED on the front of the card. The status of the parity circuit can be obtained by issuing an I/O read to the 7712 port address (address 39 hex) and parity can be cleared by writing to this port.

The memory on the 7712 must be initialized after power-up. A RAM test at power-up may fulfill this requirement (see section 4, Software). If a memory location is read which has not been written to since power-up, a parity error may occur.

Responses to a Parity Error

There are a number of ways to respond to a parity error. The appropriate response is dependent upon the importance of saving data in failed RAM and the importance of keeping the system running in spite of an error.

In many applications, it may be appropriate simply to reset the system and start over. A reset can be generated by the parity error detection circuit. If repeated attempts to restart the system fail, a hardware error can be assumed and repair would be required. Another option is to perform a complete RAM test to determine if the error is a hardware error. The RAM test can indicate which card and even which chip failed. If it is important for the system to keep running, the processor can switch to the use of a backup card. If it is important not to lose any data, a backup system could be used. This can consist of redundant 7712 cards in a system, or a redundant system. If redundant 7712 cards within a system are used, the total number is still limited to 512K bytes in 8-bit systems or 1 megabyte in 16-bit systems. Whenever the primary system is written to, the secondary system must be written to also. Only the primary system would be read from, however. If a parity error occurs in the primary system, operation would switch to the secondary system.

For any of these responses, one method of determining if the error is an isolated incident or a component failure, is for the response software to count the number of errors which have occurred with respect to elapsed time. If the error is an isolated incident, the system may be able to continue.

Responses to parity errors are jumper-selectable. The user has three options:

No Reponse

By writing to the 7712 I/O port (39 hex as shipped), the parity error is cleared. Removing all jumpers from jumper group W2 (figure 2-3, page 2-4) also disables the detection circuit output. If an error occurs on instruction fetch, it causes the wrong instruction to be executed; if an error occurs on data, the wrong data is used. This is no different from what might happen if a static RAM card or any other part of the system were to fail.

Interrupt on Parity Error

INTRQ*: The 7712 parity circuit generates an interrupt request to the system when an error occurs (as shipped configuration, W2 = 1-2). The MPU polls each 7712 for the status of its interrupt request port (data bit 0) to determine the source of the interrupt. If an I/O Read to a 7712 card indicates data bit 0 = 0, then the error was due to this card. (Each 7712 card must have a different I/O address.) Additional testing of the DRAM using the interrupt service routine could reveal which chip is faulty. The routine must not be located in DRAM memory.

NMIRQ*: By installing a jumper at W2 3-4 (figure 2-4, page 2-6), a nonmaskable interrupt can be generated when a parity error occurs. No jumper should be installed at the other positions.

The interrupt is generated on the STD BUS NMIRQ* line. The service routine could determine a hard or soft DRAM error by incrementing a failure counter (initialized at reset) and jumping to a hard error routine if a predetermined number of failures occur. If the predetermined failure number is not reached, the routine could retest the DRAM memory, report a soft error, and return to the control program. The NMIRQ* service routine should not be located in DRAM.

An alternative to the nonmaskable interrupt is to use the external PE* signal and connect it to an interrupt priority controller. This would generate a prioritized, vectored interrupt. To do this, install an RC704 cable to the 7320 priority interrupt card.

If a parity error occurs, the interrupt probably will not be recognized until the current instruction is completed. If the error occurred on the first byte of a multi-byte instruction, the processor reads the subsequent instruction bytes before the interrupt is serviced and may cause a malfunction.

The parity signal is latching and remains active until the latch is cleared. The latch is cleared by writing to the 7712 port. It may be desirable to leave the latch set since it is the same latch which lights the parity error LED on the front of the card and sets the external signal PE* active. PE* can be used to identify the faulty card in applications where the INTRQ* line is used. Feed the PE* outputs into individual inputs on a TTL I/O card for the processor to check.

Leaving the interrupt latch set should not cause repeated interrupts since nonmaskable interrupts are edge sensitive, and maskable interrupts are either edge sensitive or cannot occur again until interrupts are reenabled by the processor.

Reset On Parity Error

PBRESET*: If a system reset is necessary, both the PBRESET* and interrupt responses should be used. The interrupt will be recognized first while the reset is being debounced. The interrupt service routine should simply halt the processor so that it does not perform any further action until the reset is debounced and a system reset is generated.

Debounced Reset: To cause both a reset and an interrupt to be generated on parity error, install jumpers at W2, 1-2 and 5-6. Considerations for use of the interrupt are covered under the Interrupt On Parity Error heading. Only the use of the reset on parity error is covered here.

When the parity error occurs, it forces the STD BUS PBRESET* line low. This line is the reset input to the processor card. The processor card must respond by generating a system reset pulse on the STD BUS SYSRESET* line (all Pro-Log processor cards do so). When the reset pulse is generated on the SYSRESET* line, the parity error latch is cleared releasing PBRESET*. Note that SYSRESET* also turns off the parity error LED on the front of the card and clears the parity error output signal, PE* on connector J2 pin 2. Therefore, if if it was a hardware error and more than one 7712 card is used in the system, testing must be performed to determine the faulty card.

At reset, the 7712 clears its block registers and parity latch (the 7712 continues refreshing during all processor states!). The reset returns the processor to the bootstrap or other memory it would normally be in at power-up or reset. This is desirable since the parity error response software should not be in dynamic memory anyway as it may be this very memory which is failing. If the system automatically tries to restart itself after reset and if the error was a hardware error, the system could get caught in an endless loop. (If the system restarts itself and encounters the faulty RAM, it will get reset over and over again.) One way to avoid this, if the application permits, is to present an operator with the option of selecting normal operation or a RAM test.

Nondebounced Reset On Parity Error: If the system does not debounce the PBRESET* line, the parity error circuit can be configured to generate reset only. To generate a reset only on parity error, install jumper W2 5-6, but not 1-2. All other considerations for the use of reset described under the previous heading apply.

REFRESH

The 7712 onboard refresh circuit operation is completely transparent to the user. Since refresh occurs during all modes of processor and DMA control, the 7712 appears to be static memory. The user need only initialize all parity bits at power-up (write any data to all locations). Each of the DRAM's 256 row addresses is refreshed every 13 microseconds for a total refresh period of 3.33 milliseconds (this is called distributed refresh). The 7712 has three refresh modes:

Asynchronous — Occurs when there is no activity on the STD BUS (all lines are static). No bus activity occurs during reset (all lines are in their inactive states) and extended wait periods (some lines are in their active states). The WAITRQ* line is active during asynchronous refresh. There is no degradation of system performance since there is no bus activity. The WAITRQ* line is never used when the bus is active.

Synchronous — Occurs when there is bus activity (any read or write) without accessing DRAM memory.

Forced — Occurs during continuous accesses to DRAM memory.

The refresh circuitry consists of a digital refresh timer, a refresh enable latch and the DRAM controller. The controller monitors the refresh timer and the refresh enable outputs to determine which mode of refresh will occur. The timer is a free running oscillator (period of 13 microseconds), which sets the refresh enable latch on each rising edge. The enable tells the controller that a refresh is pending. The controller services the refresh (refresh enable is reset) at the end of the next access cycle on the bus (typically less than 2 microseconds) in either the synchronous or forced modes. If there is no bus activity, refresh enable remains high and the refresh timer output goes low. This condition triggers an asynchronous refresh from the controller and the refresh enable latch is reset (see figure 3-15).

The DRAM controller automatically determines the refresh mode during DMA as well as processor operations. These advanced refresh features permit hardware single stepping of DRAM programs, continuous DMA accesses of DRAM memory, and data retention during reset or extended wait periods without any software overhead or processor performance degradation. There are no jumpers to configure for refresh.

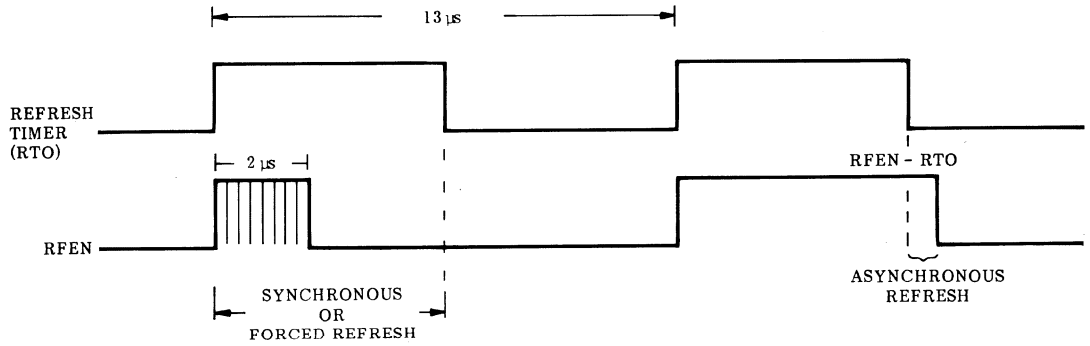


Figure 3-15. Refresh Modes

DMA PROCESSOR COMPATIBILITY

Hardware — The 7712 operates with any DMA processor which supplies address, RD*, WR*, and MEMRQ* signals, and is compatibile with the timing specifications shown on pages 2-10, 2-11, and 2-12. The most critical timing parameter is t_{c1} during forced refresh (847 nanoseconds).

Software — When using a 16-bit DMA processor in a 20-bit 8088 system (W3 1-2), the 8088 Segment Mode Register has two functions. The lower half generates address lines A19-A16 during normal processor operations (BUSAK*=1) and the upper half generates A19-A16 during DMA operations. Therefore, the upper half of this register (data bits D7-D4) should be initialized (I/O Write to Port 39) with the source/destination block needed during the DMA transfer. See 8088 Segment Mode Select and figure 3-5, page 3-6, for more information.

Section 4

OPERATING SOFTWARE

INTRODUCTION

This section contains 8085 hardware-level subroutine modules with which to operate the 7712 card. The software in this section can be used without license from Pro-Log. Although tested and believed correct, this software is not represented to be free from errors or copyright infringement, or appropriate for any specific application.

The subroutines are in STD instruction mnemonics. The coding forms are grouped at the end of this section, following the flowcharts and subroutine specifications.

MEMORY ADDRESSES

Full memory addresses are given. They are preferred addresses that allow the subroutines to work with those provided for other Series 7000 STD BUS cards from Pro-Log. The program addresses correspond to the 7712 onboard memory sockets.

If your system cannot use the memory addresses in the 7712 software package, simply change the memory page addresses, as required, when loading these modules into your system.

I/O PORT ADDRESSES

The 7712 I/O ports are assigned preferred hexadecimal address 39 for compatibility with other Series 7000 STD BUS cards. Section 2 shows how to remap this address if necessary. This software can be used by simply changing the port addresses when loading the program modules into your system.

JUMPER CONFIGURATIONS

These routines assume a 512K 7712 memory array mapped into the 8085 upper 32K address range. See basic arrangement 1, page 3-7. The jumpers configure the 7712 for the 8085 32K block mode, i.e., W4 1-3, 2-4; W5 3-4; W8 3-4; and W9 all jumpers in place. The (DRAM TEST) subroutine is located in the 8085 address range 2100-21FF. The parity error circuit is not used for these routines.

SUBROUTINES PROVIDED

These subroutines provide aid in testing and managing the memory on the 7712. Figure 4-1 is a list of the routines and their entry requirements and addresses for reference. Complete specifications are given in the following pages.

Some of the modules are interdependent. That is, one module may use part of other modules. Read the functional descriptions of each module carefully. It would probably be advantageous to load these modules into PROM and plug them into permanently enabled memory sockets in your system.

NAME AND ENTRY ADDRESS	DESCRIPTION	ENTRY REQUIRE-MENTS
(DRAM TEST)	Tests dynamic RAM.	None
(OUT BLOCK)	Outputs contents of register B to 7712 Block Register.	Address 2101H reflects initial state of block pointer port. Reg. B contains current block number.
(INIT PAGE)	Initializes all 512K locations to their page address.	Address 2101H reflects initial state of block pointer port. Reg. B contains current block number.
(COMP PAGE)	Compares all memory to page address.	None
(INIT DATA)	Writes same data to all 512K locations.	Address 2100H contains data.
(COMP DATA)	Compares all memory to data.	Address 2100H contains data.
(WALKING ONE)	Writes and compares a walking one data pattern to all 512K locations.	None

Figure 4-1. List of Subroutines (7712)

SUBROUTINE SPECIFICATIONS (DRAM TEST)

ENTRY ADDRESS	:	2102
REGISTERS ALTERED	:	A,B,C,D,E,H,L
SUBROUTINES USED	:	(INIT PAGE), (COMP PAGE), (INIT DATA), (COMP DATA), (WALKING ONE), (OUT BLOCK)
EXECUTION TIME	:	
8085A	:	Approximately 100 seconds per 512K card using a 6.25 MHz crystal
ENTRY REQUIREMENTS	:	None
EXIT CONDITIONS	:	An error should jump to a user DRAM Error routine. The 7712 RAM block is enabled in the upper 32K address range.

FUNCTIONAL DESCRIPTION

This module performs a test of the RAM on all 7712s in an 8-bit system. The program itself resides in the lower 32K address range and requires that there be static RAM available in the lower 32K. The static RAM and whatever memory this program is stored in must both be either permanently enabled or be enabled with MEMEX low.

The RAM test first writes to all the DRAM, loading each byte with the equivalent of that byte's page address (the upper eight address bits). Next, each byte is checked to see that it has the correct data in it. Next, the DRAM is loaded with all zeros (may be any data) and again checked for correct data. Finally, a one is "walked" through every location and each location is checked. If an error occurs, "hooks" are provided for an unconditional jump to the user's DRAM Error subroutine at locations 0124, 0160, and 01A4 hex. At the time of an error, the registers contain the following:

Register B contains the present DRAM block number
Register C is "don't care"
Register D contains the write data (correct data)
Register E contains the read data (incorrect data)
Register HL contains the address of the error

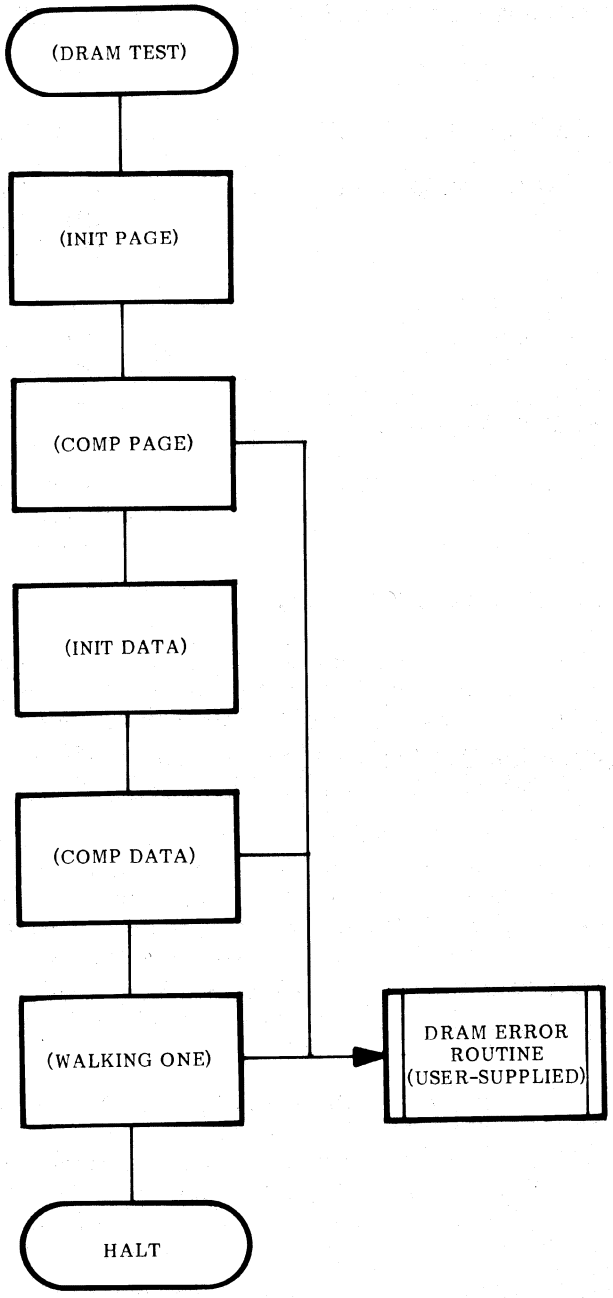


Figure 4-2. (DRAM TEST) Flowchart (7712)

SUBROUTINE SPECIFICATIONS (INIT PAGE)

ENTRY ADDRESS	:	2130
REGISTERS ALTERED	:	A,B,H,L
SUBROUTINES USED	:	(OUT BLOCK)
EXECUTION TIME 8085A	:	7 seconds
ENTRY REQUIREMENTS	:	Number of 32K memory blocks in address 2101.
EXIT CONDITIONS	:	Memory = page address in all 512K of memory.

FUNCTIONAL DESCRIPTION

The (INIT PAGE) subroutine writes to all DRAM. Each byte of memory is loaded with its respective page address. The subroutine loops until the page addresses are written to all 32K locations in all 16 blocks of memory.

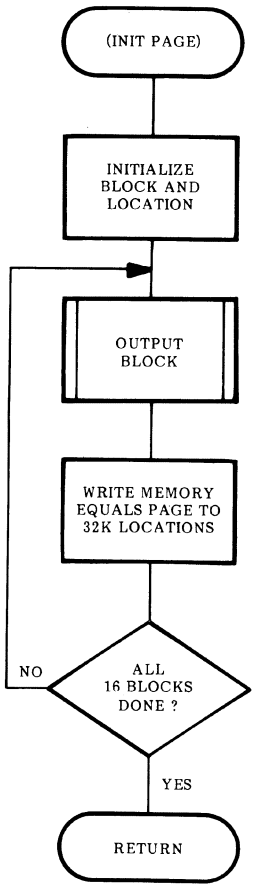


Figure 4-3. (INIT PAGE) Flowchart (7712)

SUBROUTINE SPECIFICATIONS (COMP PAGE)

ENTRY ADDRESS	:	2150
REGISTERS ALTERED	:	A,B,D,E,H,L
SUBROUTINES USED	:	(OUT BLOCK)
EXECUTION TIME 8085A	:	9 seconds
ENTRY REQUIREMENTS	:	Number of 32K memory blocks in address 2101. Memory = page address in memory blocks.
EXIT CONDITIONS	:	A DRAM error causes a jump to a user-supplied error routine.

FUNCTIONAL DESCRIPTION

This subroutine checks each DRAM location to see if it has the correct data. If an error is detected, a user-supplied DRAM Error routine should be provided.

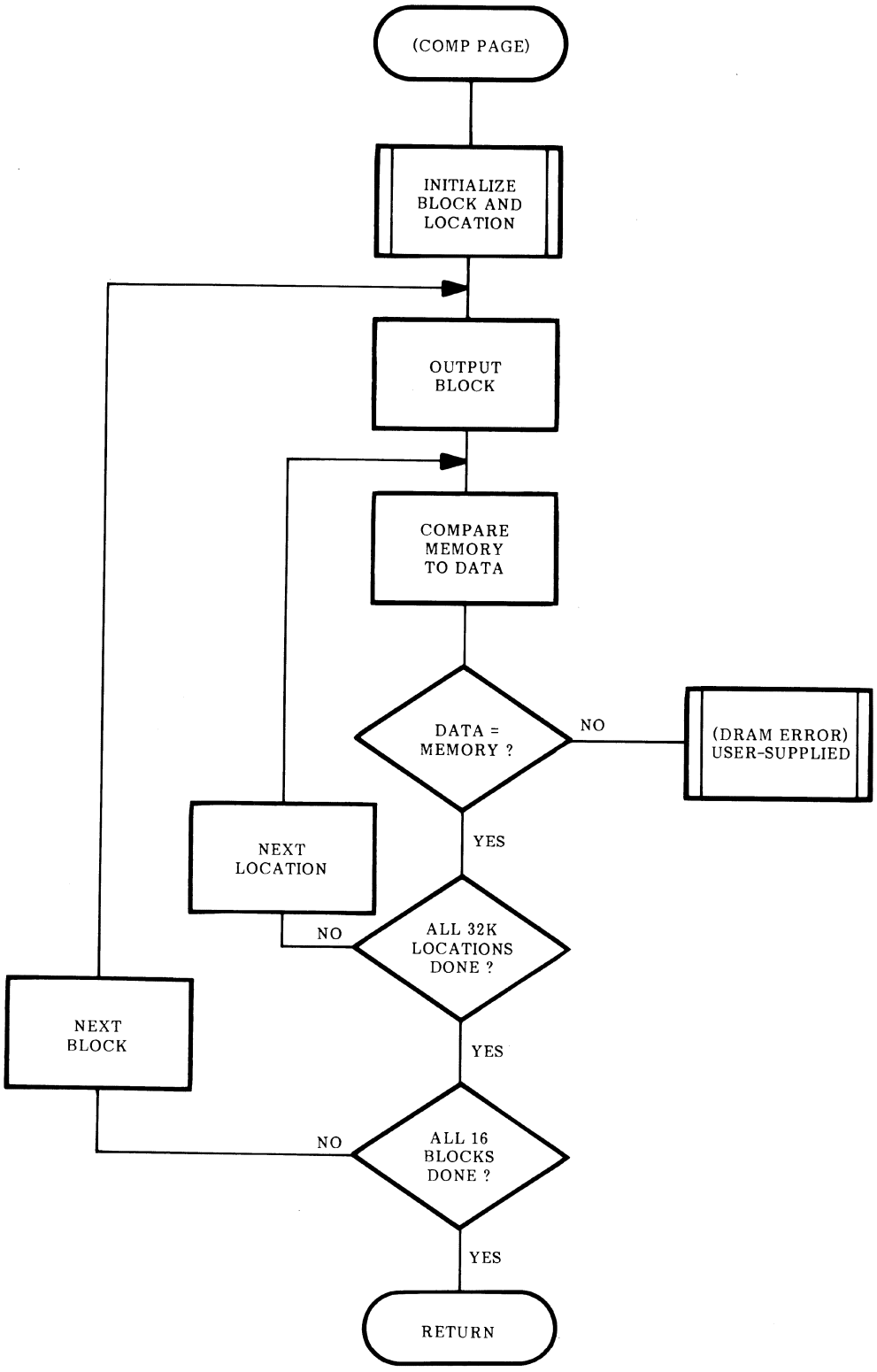


Figure 4-4. (COMP PAGE) Flowchart (7712)

SUBROUTINE SPECIFICATIONS (INIT DATA)

ENTRY ADDRESS	:	2170
REGISTERS ALTERED	:	A,B,H,L
SUBROUTINES USED	:	(OUT BLOCK)
EXECUTION TIME 8085A	:	7 seconds
ENTRY REQUIREMENTS	:	Number of 32K memory blocks in address 2101. Data byte to be written to memory is stored in address 2100.
EXIT CONDITIONS	:	Memory = data in all 512K of memory.

FUNCTIONAL DESCRIPTION

The (INIT DATA) subroutine writes to all 512K bytes of DRAM. Each memory location is loaded with a specified data byte. This subroutine loops until the data is written to all 32K locations in all 16 blocks of memory.

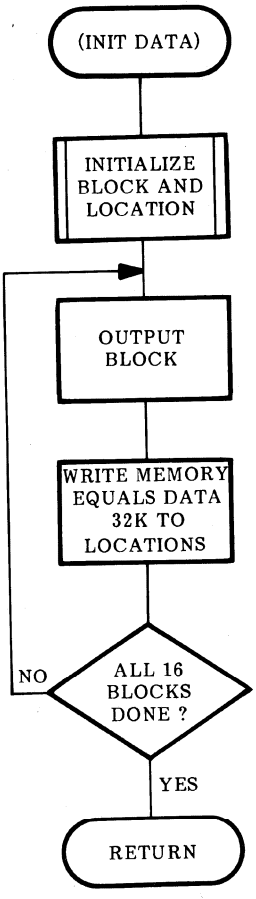


Figure 4-5. (INIT DATA) Flowchart (7712)

SUBROUTINE SPECIFICATIONS (COMP DATA)

ENTRY ADDRESS	:	2190
REGISTERS ALTERED	:	A,B,E,H,L
SUBROUTINES USED	:	(OUT BLOCK)
EXECUTION TIME 8085A	:	9 seconds
ENTRY REQUIREMENTS	:	Number of 32K memory blocks in address 2101. Data byte to be written to memory is stored in address 2100.
EXIT CONDITIONS	:	A DRAM error causes a jump to a user-supplied error routine.

FUNCTIONAL DESCRIPTION

This subroutine checks each DRAM location to see if it has the correct data in it. If an error is detected, a user-supplied DRAM Error routine should be provided.

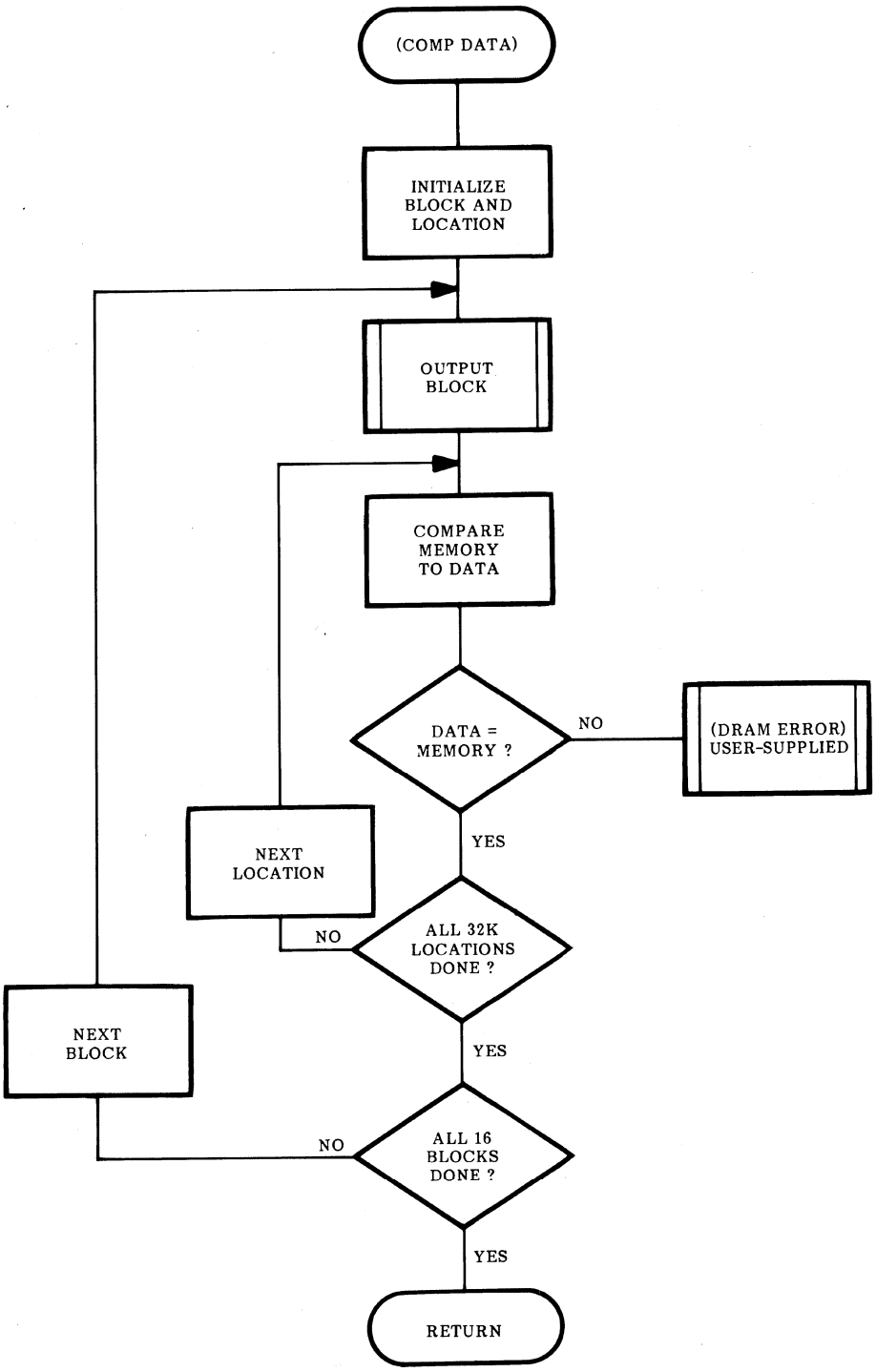


Figure 4-6. (COMP DATA) Flowchart (7712)

SUBROUTINE SPECIFICATIONS (WALKING ONE)

ENTRY ADDRESS	:	21B0
REGISTERS ALTERED	:	A,B,D,E,H,L
SUBROUTINES USED	:	(OUT BLOCK)
EXECUTION TIME 8085A	:	68 seconds
ENTRY REQUIREMENTS	:	Number of 32K memory blocks in address 2101.
EXIT CONDITIONS	:	A DRAM error causes a jump to a user-supplied error routine.

FUNCTIONAL DESCRIPTION

This subroutine tests each bit of DRAM by rotating a one through each byte of memory. Each bit is checked as the one is "walked" through the byte. The subroutine loops until all 512K bytes of memory are checked. If a DRAM error occurs, a user-supplied error routine should be provided.

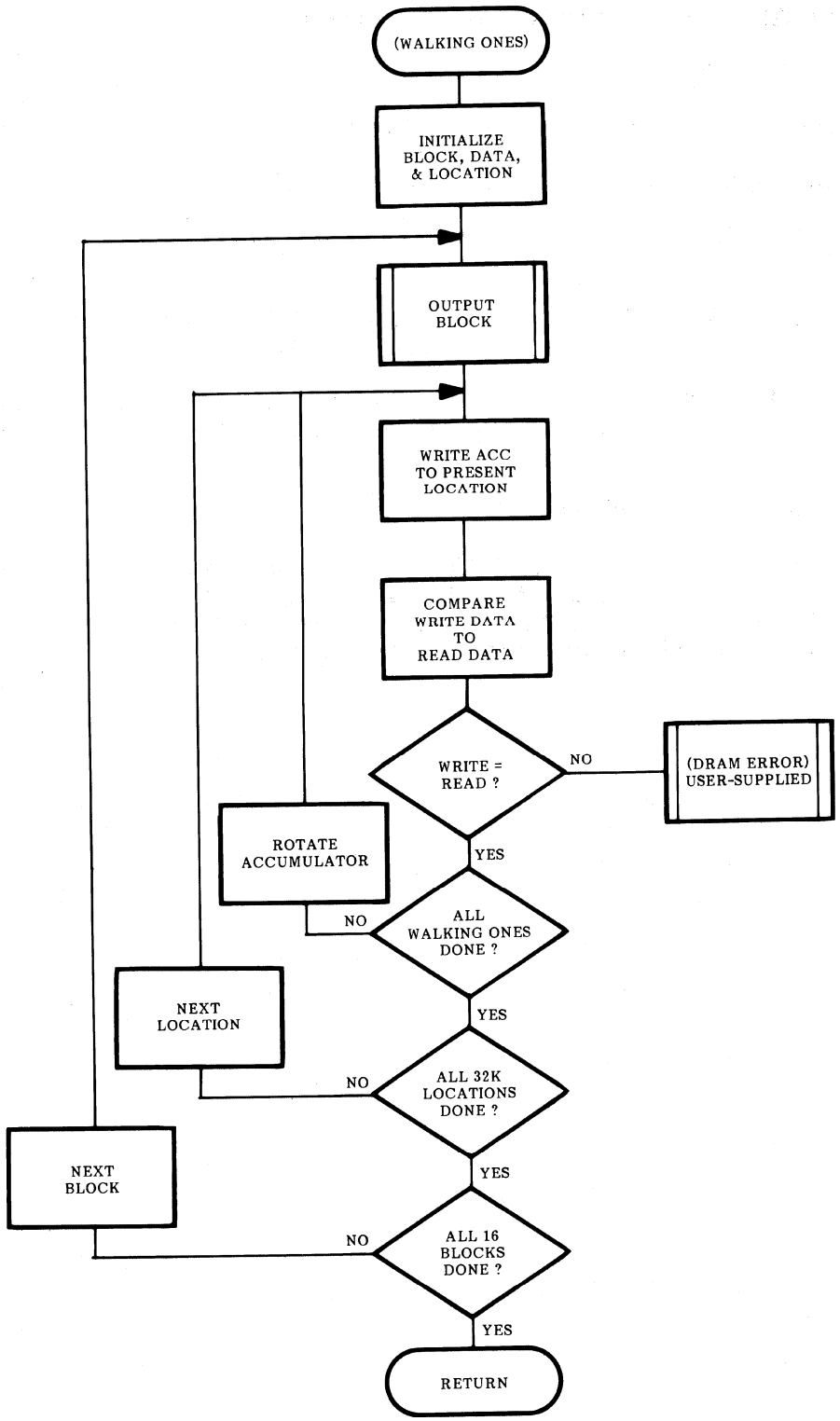


Figure 4-7. (WALKING ONE) Flowchart (7712)

SUBROUTINE SPECIFICATIONS (OUT BLOCK)

ENTRY ADDRESS	:	2120
REGISTERS ALTERED	:	A,H,L
SUBROUTINES USED	:	None
EXECUTION TIME 8085A	:	2 microseconds
ENTRY REQUIREMENTS	:	Block number to be output in Register B.
EXIT CONDITIONS	:	HL = 8000

FUNCTIONAL DESCRIPTION

The (OUT BLOCK) subroutine outputs the memory block number to the block register and initializes the memory pointer of the 32K block.

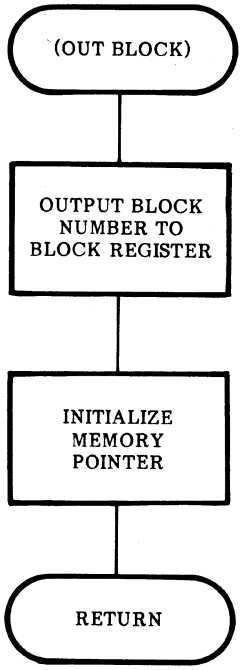


Figure 4-8. (OUT BLOCK) Flowchart (7712)

```
ADRdata!      LABEL          !INSTR !     MODIFIER     !            COMMENTS
-------------------------------------------------------------------------------
2100 00!DATA                 !  00  !                  !SET DATA = 00
2101 10!BLOCK                !  10  !                  !SET BLOCK = 10
2102 3E!DRAM TEST            !LDAI  !                  !TEST 1
  03 01!                     ! ____ !01 hex            !
  04 D3!                     !OPA   !                  !
  05 FF!                     ! ____ !FF hex            !
  06 CD!                     !JS    !UN                !INITIALIZE ENTIRE 512K
  07 30!                     ! ____ !(INIT PAGE)       !WITH DATA = PAGE ADDRESS
  08 21!                     ! ____ !                  !
  09 3E!                     !LDAI  !                  !TEST 2
  0A 02!                     ! ____ !02 hex            !
  0B D3!                     !OPA   !                  !
  0C FF!                     ! ____ !FF hex            !
  0D CD!                     !JS    !UN                !COMPARE ENTIRE MEMORY FOR
  0E 50!                     ! ____ !(COMP PAGE)       !DATA = PAGE ADDRESS
  0F 21!                     ! ____ !                  !
2110 3E!                     !LDAI  !                  !TEST 3
  11 03!                     ! ____ !03 hex            !
  12 D3!                     !OPA   !                  !
  13 FF!                     ! ____ !FF hex            !
  14 CD!                     !JS    !UN                !INITIALIZE ENTIRE MEMORY
  15 70!                     ! ____ !(INIT DATA)       !WITH DATA = 00
  16 21!                     ! ____ !                  !
  17 3E!                     !LDAI  !                  !TEST 4
  18 04!                     ! ____ !04 hex            !
  19 D3!                     !OPA   !                  !
  1A FF!                     ! ____ !FF hex            !
  1B CD!                     !JS    !UN                !COMPARE MEMORY FOR DATA = 00
  1C 90!                     ! ____ !(COMP DATA)       !
  1D 21!                     ! ____ !                  !
  1E 3E!                     !LDAI  !                  !TEST 5
  1F 05!                     ! ____ !05 hex            !
2120 D3!                     !OPA   !                  !
  21 FF!                     ! ____ !FF hex            !
  22 CD!                     !JS    !UN                !WRITE AND COMPARE ENTIRE
  23 B0!                     ! ____ !(WALKING ONE)     !MEMORY WITH A WALKING ONE
  24 21!                     ! ____ !                  !PATTERN
  25 76!   ----------        !HLT   !                  !
  26 00!                     !NOP   !                  !
  27 00!                     !NOP   !                  !
2128 78!(OUT BLOCK)          !LDA   !B                 !OUTPUT TO BLOCK REGISTER
  29 D3!                     !OPA   !                  !SUBROUTINE
  2A 39!                     ! ____ !39 hex            !
  2B 21!                     !LDPI  !HL                !IO ADDRESS = 39H
  2C 00!                     ! ____ !8000 hex          !
  2D 80!                     ! ____ !                  !
  2E C9!   ----------        !RTS   !UN                !
  2F 00!                     !NOP   !                  !
2130 3A!(INIT PAGE)          !LDAD  !                  !INITIALIZE PAGE SUBROUTINE
  31 01!                     ! ____ !BLOCK             !
  32 21!                     ! ____ !                  !
  33 47!                     !LDR   !B,A               !INITIALIZE BLOCK
2134 CD!OUT BLOCK IP         !JS    !UN                !OUTPUT BLOCK
  35 28!                     ! ____ !(OUT BLOCK)       !
  36 21!                     ! ____ !                  !
2137 7C!WRITE PAGE           !LDA   !H                 !WRITE PAGE (32K LOOPS)
  38 77!                     !STAN  !(HL)              !TO MEMORY AT PRESENT
  39 23!                     !ICP   !HL                !BLOCK
  3A 7C!                     !LDA   !H                 !
  3B B5!                     !ORA   !L                 !
  3C C2!                     !JP    !NOT ZERO          !
  3D 37!                     ! ____ !WRITE PAGE        !
  3E 21!                     ! ____ !                  !
  3F 05!                     !DCR   !B                 !
```

Figure 4-9. Software Routines (7712)

```
ADRdata!      LABEL       !INSTR !     MODIFIER     !            COMMENTS
-------------------------------------------------------------------------------
2140 C2!                  !JP    !NOT ZERO          !DO NEXT BLOCK (16 LOOPS)
  41 34!                  ! ---- !OUT BLOCK IP      !
  42 21!                  ! ---- !                  !
  43 C9!   ----------     !RTS   !UN                !ALL BLOCKS DONE
  44 FF!                  !      !                  !RETURN FROM SUBROUTINE
  45 FF!                  !      !                  !
  46 FF!                  !      !                  !
  47 FF!                  !      !                  !
  48 FF!                  !      !                  !
  49 FF!                  !      !                  !
  4A FF!                  !      !                  !
  4B FF!                  !      !                  !
  4C FF!                  !      !                  !
  4D FF!                  !      !                  !
  4E FF!                  !      !                  !
  4F FF!                  !      !                  !
2150 3A!(COMP PAGE)       !LDAD  !                  !COMPARE PAGE SUBROUTINE
  51 01!                  ! ---- !BLOCK             !
  52 21!                  ! ---- !                  !
  53 47!                  !LDR   !B,A               !INITIALIZE BLOCK
2154 CD!OUT BLOCK CP      !JS    !UN                !
  55 28!                  ! ---- !(OUT BLOCK)       !OUTPUT BLOCK
  56 21!                  ! ---- !                  !
2157 54!COMP PAGE         !LDR   !D,H               !COMPARE MEMORY TO PAGE ADDRESS
  58 7C!                  !LDA   !H                 !AT PRESENT BLOCK (32K LOOPS)
  59 5E!                  !LDRN  !E,(HL)            !
  5A BB!                  !CPA   !E                 !
  5B CA!                  !JP    !ZERO              !
  5C 62!                  ! ---- !CORRECT PAGE      !
  5D 21!                  ! ---- !                  !
215E FF!HOOK 1            !  FF  !                  !DUMP REGISTERS ON AN ERROR
  5F 00!                  !NOP   !                  !
2160 00!                  !NOP   !                  !
  61 00!                  !NOP   !                  !
2162 23!CORRECT PAGE      !ICP   !HL                !
  63 7C!                  !LDA   !H                 !
  64 B5!                  !ORA   !L                 !
  65 C2!                  !JP    !NOT ZERO          !
  66 57!                  ! ---- !COMP PAGE         !
  67 21!                  ! ---- !                  !
  68 05!                  !DCR   !B                 !
  69 C2!                  !JP    !NOT ZERO          !DO NEXT BLOCK (16 LOOPS)
  6A 54!                  ! ---- !OUT BLOCK CP      !
  6B 21!                  ! ---- !                  !
  6C C9!   ----------     !RTS   !UN                !ALL BLOCKS DONE, RETURN FROM
  6D FF!                  !      !                  !SUBROUTINE
  6E FF!                  !      !                  !
  6F FF!                  !      !                  !
2170 3A!(INIT DATA)       !LDAD  !                  !(INITIALIZE DATA SUBROUTINE)
  71 01!                  ! ---- !BLOCK             !
  72 21!                  ! ---- !                  !
  73 47!                  !LDR   !B,A               !INITIALIZE BLOCK
2174 CD!OUT BLOCK ID      !JS    !UN                !OUTPUT BLOCK
  75 28!                  ! ---- !(OUT BLOCK)       !
  76 21!                  ! ---- !                  !
2177 3A!WRITE DATA        !LDAD  !                  !WRITE DATA TO MEMORY AT
  78 00!                  ! ---- !DATA              !PRESENT BLOCK (32K LOOPS)
  79 21!                  ! ---- !                  !
  7A 77!                  !STAN  !(HL)              !
  7B 23!                  !ICP   !HL                !
  7C 7C!                  !LDA   !H                 !
  7D B5!                  !ORA   !L                 !
  7E C2!                  !JP    !NOT ZERO          !
  7F 77!                  ! ---- !WRITE DATA        !
```

Figure 4-9. Software Routines (Continued)

```
ADRdata!      LABEL        !INSTR !    MODIFIER    !          COMMENTS
-------------------------------------------------------------------------------
2180 21!                   ! ---- !                !
  81 05!                   !DCR   !B               !
  82 C2!                   !JP    !NOT ZERO        !DO NEXT BLOCK
  83 74!                   ! ---- !OUT BLOCK ID    !
  84 21!                   ! ---- !                !
  85 C9!   ----------      !RTS   !UN              !ALL BLOCKS DONE, RETURN
  86 FF!                   !      !                !FROM SUBROUTINE
  87 FF!                   !      !                !
  88 FF!                   !      !                !
  89 FF!                   !      !                !
  8A FF!                   !      !                !
  8B FF!                   !      !                !
  8C FF!                   !      !                !
  8D FF!                   !      !                !
  8E FF!                   !      !                !
  8F FF!                   !      !                !
2190 3A!(COMP DATA)        !LDAD  !                !(COMPARE DATA SUBROUTINE)
  91 01!                   ! ---- !BLOCK           !INITIALIZE FIRST BLOCK
  92 21!                   ! ---- !                !
  93 47!                   !LDR   !B,A             !
2194 CD!OUT BLOCK CD       !JS    !UN              !OUTPUT BLOCK
  95 28!                   ! ---- !(OUT BLOCK)     !
  96 21!                   ! ---- !                !
2197 3A!COMP DATA          !LDAD  !                !COMPARE DATA TO MEMORY
  98 00!                   ! ---- !DATA            !AT PRESENT BLOCK (32K LOOPS)
  99 21!                   ! ---- !                !
  9A 57!                   !LDR   !D,A             !
  9B 5E!                   !LDRN  !E,(HL)          !
  9C BB!                   !CPA   !E               !
  9D CA!                   !JP    !ZERO            !
  9E A4!                   ! ---- !CORRECT DATA    !
  9F 21!                   ! ---- !                !
21A0 FF!HOOK 2             !  FF  !                !DUMP REGISTERS ON AN ERROR
  A1 00!                   !NOP   !                !
  A2 00!                   !NOP   !                !
  A3 00!                   !NOP   !                !
21A4 23!CORRECT DATA       !ICP   !HL              !
  A5 7C!                   !LDA   !H               !
  A6 B5!                   !ORA   !L               !
  A7 C2!                   !JP    !NOT ZERO        !
  A8 97!                   ! ---- !COMP DATA       !
  A9 21!                   ! ---- !                !
  AA 05!                   !DCR   !B               !
  AB C2!                   !JP    !NOT ZERO        !DO NEXT BLOCK (16 LOOPS)
  AC 94!                   ! ---- !OUT BLOCK CD    !
  AD 21!                   ! ---- !                !
  AE C9!   ----------      !RTS   !UN              !ALL BLOCKS DONE, RETURN
  AF FF!                   !      !                !FROM SUBROUTINE
21B0 3A!(WALKING ONE)      !LDAD  !                !(WALKING ONE SUBROUTINE)
  B1 01!                   ! ---- !BLOCK           !INITIALIZE BLOCK
  B2 21!                   ! ---- !                !
  B3 47!                   !LDR   !B,A             !
21B4 CD!OUT BLOCK WO       !JS    !UN              !OUTPUT BLOCK
  B5 28!                   ! ---- !(OUT BLOCK)     !
  B6 21!                   ! ---- !                !
21B7 AF!WRITE ONE          !XRA   !A               !INTIALIZE ACCUMULATOR
  B8 3E!                   !LDAI  !                !AND CARRY
  B9 01!                   ! ---- !01 hex          !
21BA 57!NEXT DATA          !LDR   !D,A             !STORE AND COMPARE WALKING ONE
  BB 72!                   !STRN  !D,(HL)          !(8 LOOPS) TO PRESENT MEMORY
  BC 5E!                   !LDRN  !E,(HL)          !LOCATION
  BD BB!                   !CPA   !E               !
  BE CA!                   !JP    !ZERO            !
  BF C5!                   ! ---- !CORRECT ONE     !
```

Figure 4-9. Software Routines (continued)

```
ADRdata!      LABEL       !INSTR !     MODIFIER     !            COMMENTS
-------------------------------------------------------------------------------
21C0 21!                  ! ---- !                  !
21C1 FF!HOOK 3            !  FF  !                  !DUMP REGISTERS ON ERROR
  C2 00!                  !NOP   !                  !
  C3 00!                  !NOP   !                  !
  C4 B7!                  !ORA   !A                 !CLEAR CARRY ON AN ERROR
21C5 17!CORRECT ONE       !RLAC  !                  !
  C6 D2!                  !JP    !NO CARRY          !
  C7 BA!                  ! ---- !NEXT DATA         !
  C8 21!                  ! ---- !                  !
  C9 23!                  !ICP   !HL                !DO NEXT MEMORY LOCATION
  CA 7C!                  !LDA   !H                 !(32K LOOPS)
  CB B5!                  !ORA   !L                 !
  CC C2!                  !JP    !NOT ZERO          !
  CD B7!                  ! ---- !WRITE ONE         !
  CE 21!                  ! ---- !                  !
  CF 05!                  !DCR   !B                 !
21D0 C2!                  !JP    !NOT ZERO          !DO NEXT BLOCK (16 LOOPS)
  D1 B4!                  ! ---- !OUT BLOCK WO      !
  D2 21!                  ! ---- !                  !
  D3 C9!   ----------     !RTS   !UN                !ALL BLOCK DONE, RETURN FROM
  D4 FF!                  !      !                  !SUBROUTINE
  D5 FF!                  !      !                  !
  D6 FF!                  !      !                  !
  D7 FF!                  !      !                  !
  D8 FF!                  !      !                  !
  D9 FF!                  !      !                  !
  DA FF!                  !      !                  !
  DB FF!                  !      !                  !
  DC FF!                  !      !                  !
  DD FF!                  !      !                  !
  DE FF!                  !      !                  !
  DF FF!                  !      !                  !
21E0 FF!                  !      !                  !
  E1 FF!                  !      !                  !
  E2 FF!                  !      !                  !
  E3 FF!                  !      !                  !
  E4 FF!                  !      !                  !
  E5 FF!                  !      !                  !
  E6 FF!                  !      !                  !
  E7 FF!                  !      !                  !
  E8 FF!                  !      !                  !
  E9 FF!                  !      !                  !
  EA FF!                  !      !                  !
  EB FF!                  !      !                  !
  EC FF!                  !      !                  !
  ED FF!                  !      !                  !
  EE FF!                  !      !                  !
  EF FF!                  !      !                  !
21F0 FF!                  !      !                  !
  F1 FF!                  !      !                  !
  F2 FF!                  !      !                  !
  F3 FF!                  !      !                  !
  F4 FF!                  !      !                  !
  F5 FF!                  !      !                  !
  F6 FF!                  !      !                  !
  F7 FF!                  !      !                  !
  F8 FF!                  !      !                  !
  F9 FF!                  !      !                  !
  FA FF!                  !      !                  !
  FB FF!                  !      !                  !
  FC FF!                  !      !                  !
  FD FF!                  !      !                  !
  FE FF!                  !      !                  !
  FF FF!                  !      !                  !
```

Figure 4-9. Software Routines (continued)

REFERENCE DRAWINGS

The schematic (figure 5-1) and assembly drawing (figure 5-2) in the following pages are included in this manual FOR REFERENCE USE ONLY. These drawings may differ in some respects from the card you receive from Pro-Log. If you require the specific revision documents of the schematic and assembly for the board shipped, please use the information request form in the back of this manual to order them.

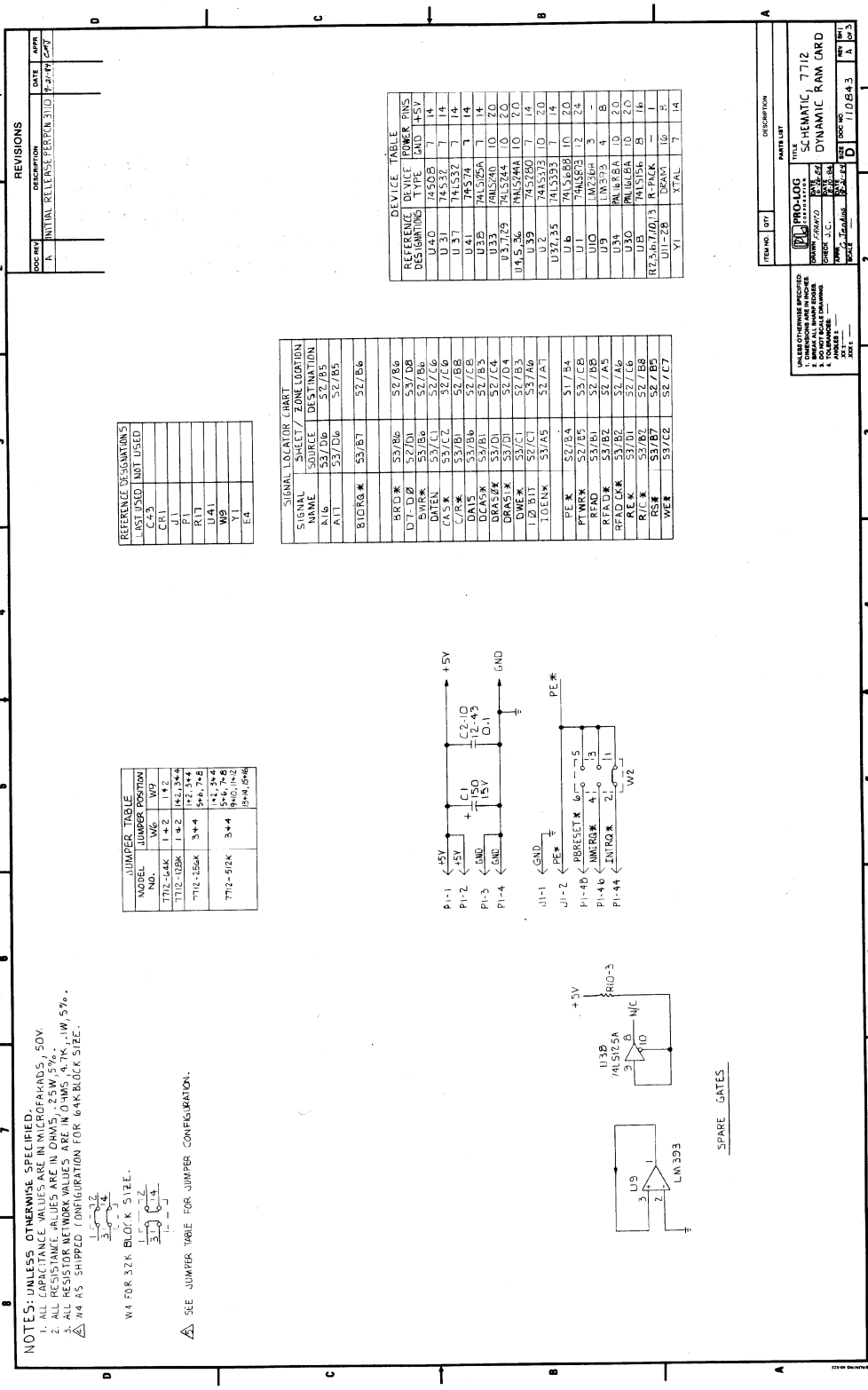

NOTES: UNLESS OTHERWISE SPECIFIED:
1. ALL CAPACITANCE VALUES ARE IN MICROFARADS, 50V
2. ALL RESISTANCE VALUES ARE IN OHMS, .25W, 5%.
3. ALL RESISTOR NETWORK VALUES ARE IN OHMS, 4.7K, .1W, 5%.
4. W4 AS SHIPPED CONFIGURATION FOR 64K BLOCK SIZE.
W4 FOR 32K BLOCK SIZE.
5. SEE JUMPER TABLE FOR JUMPER CONFIGURATION.
JUMPER TABLE
MODEL NO. | JUMPER POSITION W6 | W9
7712-64K | 1 + 2 | 1 + 2
7712-128K | 1 + 2 | 1+2, 3+4
7712-256K | 3+4 | 1+2, 3+4, 5+6, 7+8
7712-512K | 3+4 | 1+2, 3+4, 5+6, 7+8, 9+10, 11+12, 13+14, 15+16
REFERENCE DESIGNATIONS
MUST USED | NOT USED
C43
CR1
J1
P1
R17
U41
W9
Y1
E4
SIGNAL LOCATOR CHART
SIGNAL NAME | SHEET/ZONE LOCATION SOURCE | DESTINATION
A16 | S3/D6 | S2/B5
A17 | S3/D6 | S2/B5
BIORQ* | S3/B7 | S2/B6
BRQ* | S3/B6 | S2/B6
DT-Q0 | S2/D1 | S3/D8
BWR* | S3/B6 | S2/B6
DATEN | S3/C1 | S2/C6
CAS* | S3/C2 | S2/C6
C/R* | S3/B1 | S2/B8
DAIS | S3/B6 | S2/C8
DCAS* | S3/B1 | S2/C5
DRAS2* | S3/D1 | S2/C4
DRAS1* | S3/D1 | S2/B3
DWE* | S3/C1 | S3/A6
I/O BIT | S3/C1 | S2/A7
IOEN* | S3/A5 | S2/A7
PE* | S2/B4 | S1/B4
PTWR* | S2/B5 | S3/C8
RFRQ | S3/B1 | S2/B5
RFRQ* | S3/B2 | S2/A5
RFRQ CLK* | S3/B2 | S2/A6
R/C* | S3/D1 | S2/C6
RS* | S3/B2 | S2/B8
WE* | S3/B7 | S2/B5
S3/C2 | S2/C7
DEVICE TABLE
REFERENCE DESIGNATION | DEVICE TYPE | POWER PINS GND | +5V
U40 | 1450B | 7 | 14
U31 | 74S32 | 7 | 14
U37 | 74LS32 | 7 | 14
U41 | 74S74 | 7 | 14
U38 | 74LS125A | 7 | 14
U33 | 74LS240 | 10 | 20
U3,7,29 | 74LS244 | 10 | 20
U4,5,26 | 74ALS244A | 10 | 20
U39 | 74S280 | 7 | 14
U2 | 74ALS273 | 10 | 20
U32,35 | 74LS393 | 7 | 14
U6 | 74LS688 | 10 | 20
U1 | 74ALS623 | 12 | 24
U10 | LM2404 | 3 | -
U9 | LM393 | 4 | 8
U34 | PAL16R8A | 10 | 20
U30 | PAL16L8A | 10 | 20
U8 | 74LS156 | 8 | 16
RZ3,6,7,10,13 | R-PACK | — | 1
U11-28 | DRAM | 16 | 8
Y1 | XTAL | 7 | 14
+5V
GND
PE*
PBRESET*
WAITRQ*
INTRQ*
C1 150 15V
C2-10 12-43 0.1
P1-1
P1-2
P1-3
P1-4
P1-48
P1-46
P1-44
J1-1
J1-2
W2
R10-3
U38 74LS125A
U9 LM393
N/C
SPARE GATES
REVISIONS
DOC REV | DESCRIPTION | DATE | APPR
A | INITIAL RELEASE PER PCN 3110 | 8-21-89 | CMT
PRO-LOG
SCHEMATIC, 7712 DYNAMIC RAM CARD
DOC NO 110843
REV A
SHT 1 OF 3

Figure 5-1. 7712 Schematic, Page 1 of 3

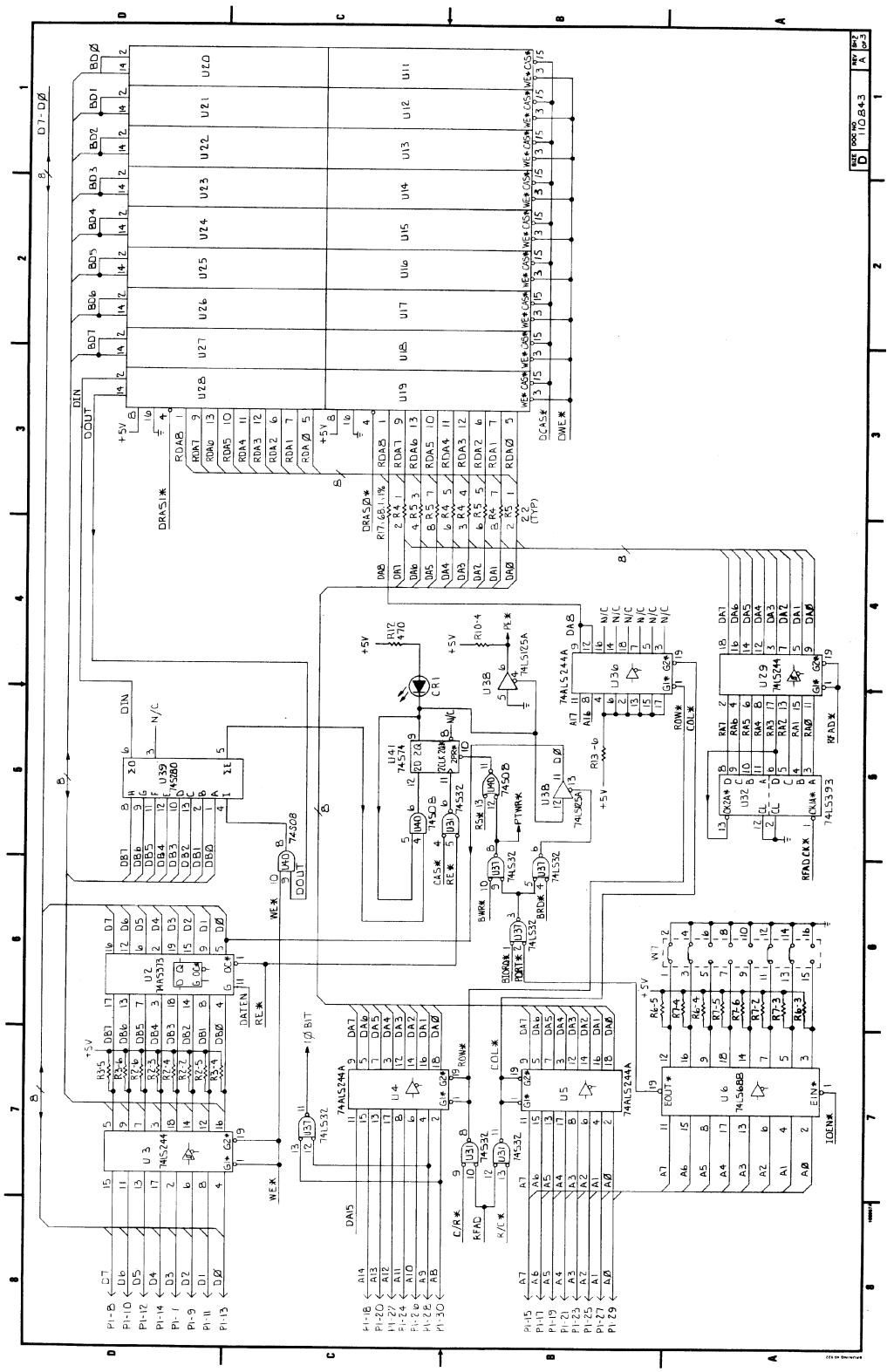


Figure 5-1. 7712 Schematic, Page 2 of 3

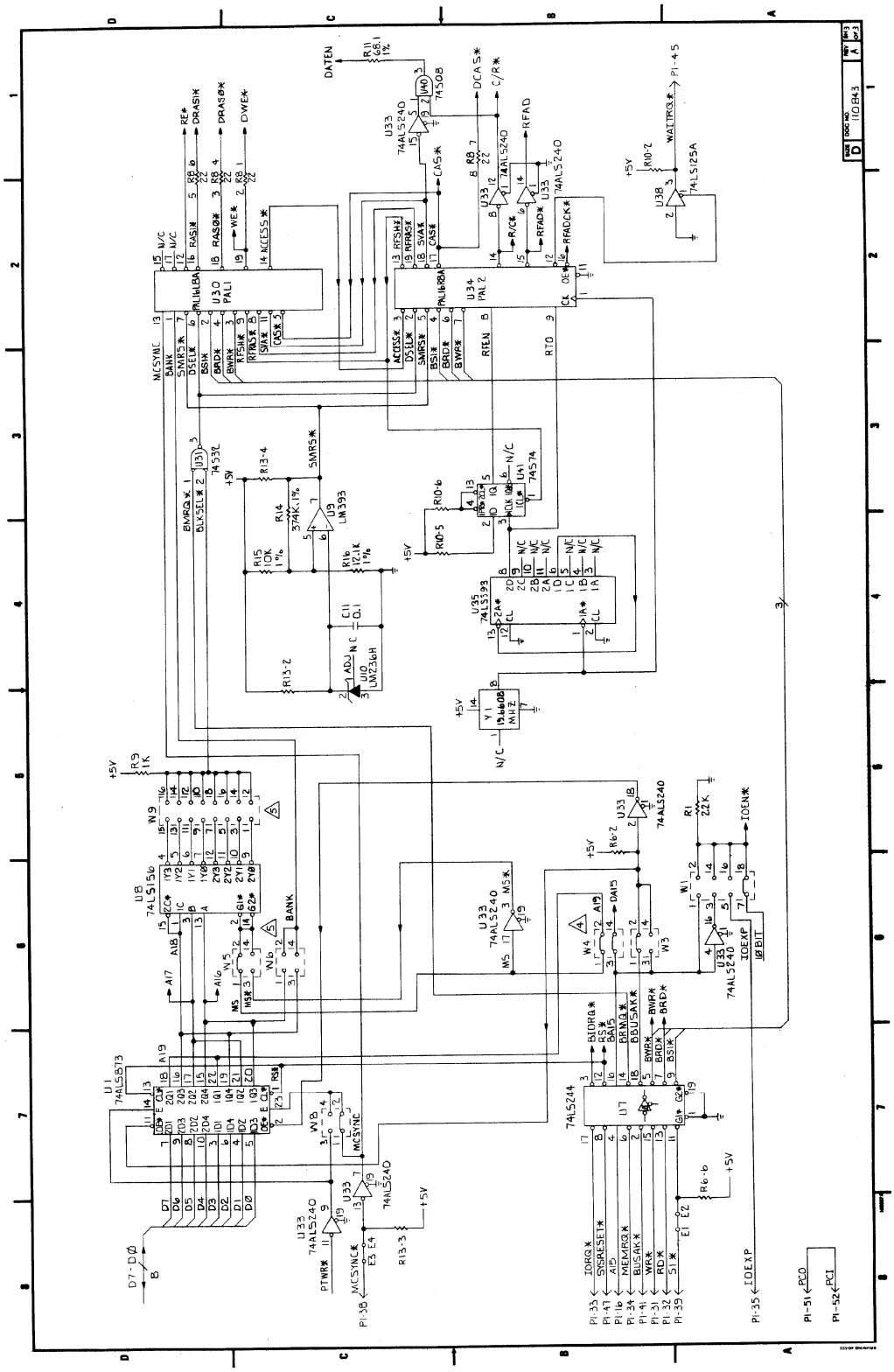
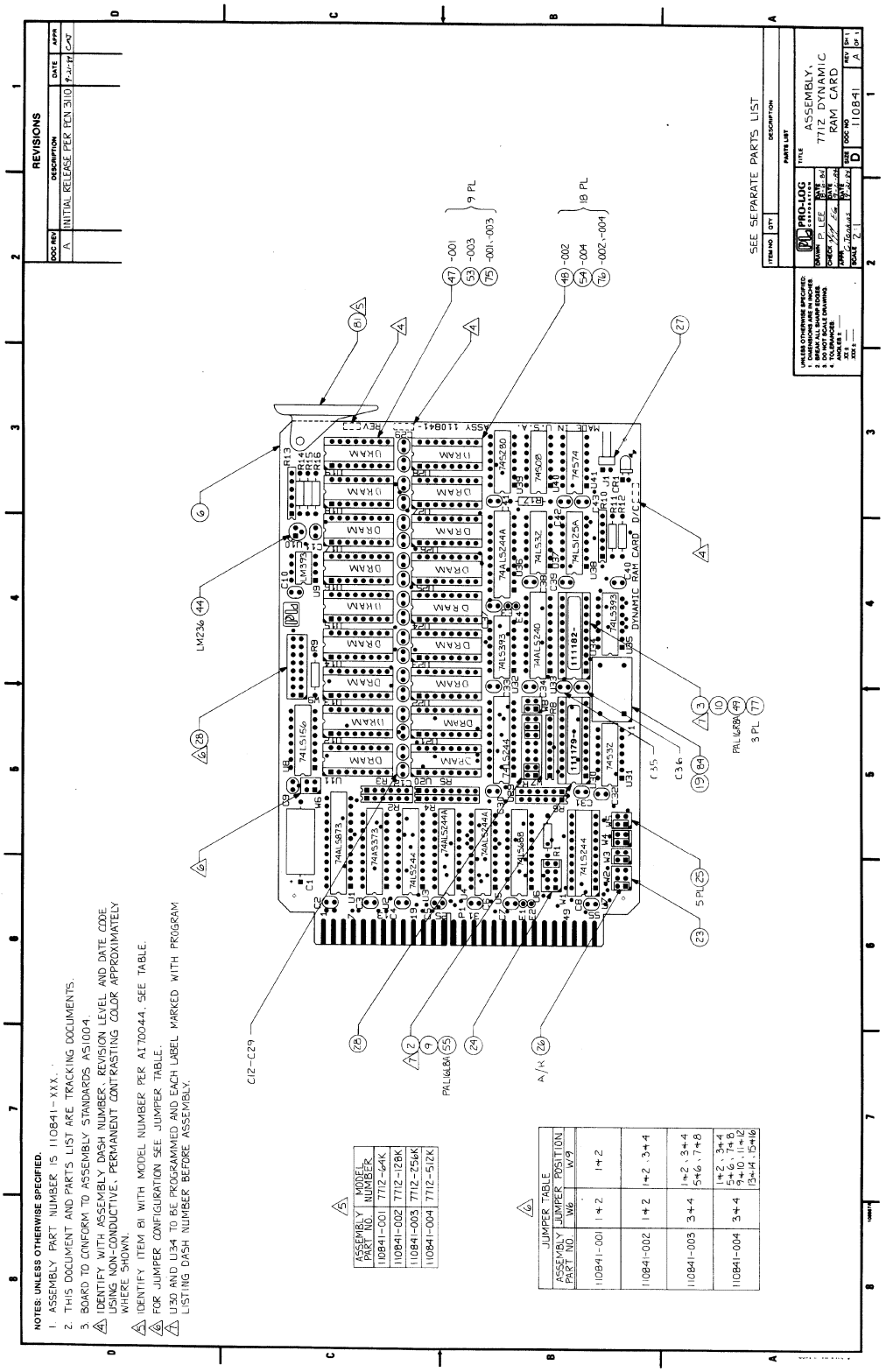


Figure 5-1. 7712 Schematic, Page 3 of 3


REVISIONS
DOC NO
DESCRIPTION
DATE
APPR
A
INITIAL RELEASE PER PEN 3110
4-21-87
NOTES: UNLESS OTHERWISE SPECIFIED.
1. ASSEMBLY PART NUMBER IS 110841-XXX.
2. THIS DOCUMENT AND PARTS LIST ARE TRACKING DOCUMENTS.
3. BOARD TO CONFORM TO ASSEMBLY STANDARDS AS1004.
4 IDENTIFY WITH ASSEMBLY DASH NUMBER, REVISION LEVEL AND DATE CODE USING NON-CONDUCTIVE, PERMANENT CONTRASTING COLOR APPROXIMATELY WHERE SHOWN.
5 IDENTIFY ITEM 81 WITH MODEL NUMBER PER A170044, SEE TABLE.
6 FOR JUMPER CONFIGURATION SEE JUMPER TABLE.
7 U30 AND U34 TO BE PROGRAMMED AND EACH LABEL MARKED WITH PROGRAM LISTING DASH NUMBER BEFORE ASSEMBLY.
ASSEMBLY PART NO | MODEL NUMBER
110841-001 | 7712-64K
110841-002 | 7712-128K
110841-003 | 7712-256K
110841-004 | 7712-512K
JUMPER TABLE
ASSEMBLY PART NO | JUMPER W6 | W9
110841-001 | 1+2 | 1+2
110841-002 | 1+2 | 1+2, 3+4
110841-003 | 3+4 | 1+2, 3+4 5+6, 7+8
110841-004 | 3+4 | 1+2, 3+4 5+6, 7+8 9+10, 11+12 13+14, 15+16
SEE SEPARATE PARTS LIST
ITEM NO
QTY
DESCRIPTION
PARTS LIST
PRO-LOG
DRAWN P. LEE
CHECKED
APPR
SCALE 2:1
TITLE
ASSEMBLY, 7712 DYNAMIC RAM CARD
SIZE D
DOC NO 110841
REV A
SHT 1 OF 1
DYNAMIC RAM CARD
ASSY 110841-
LM236
LM336
74LS373
74LS244
74LS244A
74LS245A
74LS532
74LS393
74LS240
74LS86
74LS08
74LS74
74LS125A
74LS32
74LS00
74LS04
74LS273
74LS157
74LS158
74LS283
74LS85
74LS51
74LS10
74LS20
74LS27
74LS30
74LS42
74LS123
74LS138
74LS139
74LS151
74LS153
74LS155
74LS156
74LS161
74LS163
74LS164
74LS165
74LS166
74LS169
74LS174
74LS175
74LS181
74LS190
74LS191
74LS192
74LS193
74LS194
74LS195
74LS221
74LS240
74LS241
74LS242
74LS243
74LS245
74LS247
74LS248
74LS249
74LS251
74LS253
74LS257
74LS258
74LS259
74LS260
74LS266
74LS273
74LS280
74LS283
74LS290
74LS293
74LS298
74LS299
74LS322
74LS323
74LS352
74LS353
74LS365
74LS366
74LS367
74LS368
74LS373
74LS374
74LS377
74LS378
74LS379
74LS381
74LS382
74LS385
74LS386
74LS390
74LS393
74LS395
74LS396
74LS398
74LS399
74LS424
74LS425
74LS426
74LS428
74LS433
74LS436
74LS437
74LS438
74LS439
74LS440
74LS441
74LS442
74LS443
74LS444
74LS445
74LS446
74LS447
74LS448
74LS449
74LS450
74LS451
74LS452
74LS453
74LS454
74LS455
74LS456
74LS457
74LS458
74LS459
74LS460
74LS461
74LS462
74LS463
74LS464
74LS465
74LS466
74LS467
74LS468
74LS469
74LS470
74LS471
74LS472
74LS473
74LS474
74LS475

RETURN FOR REPAIR PROCEDURES

Domestic Customers:

1. Call CUSTOMER SERVICE at (408) 372-4593 (in California) or (800) 538-9570 (outside of California) 7 AM to 3 PM PST.

2. Explain the problem and we may be able to solve it on the phone. If not, we will give you a Customer Return Order (CRO) number.

   Mark the CRO number on the shipping label, packing slip, and other paperwork accompanying the return. We cannot accept returns without a CRO.

3. Please be sure to enclose a packing slip with the CRO number, the equipment serial number if applicable, and the reason for return. Also enclose the name and telephone number of the person we should contact (preferably the user).

4. Package the equipment in a solid cardboard box secured with packing material.

   CAUTION: Loose MOS integrated circuits, or any product containing CMOS integrated circuits, must be protected from electrostatic discharge during handling. Use conductive foam pads or conductive plastic bags, and never place MOS or CMOS circuitry in contact with Styrofoam materials.

5. Ship prepaid and insured to:

   Pro-Log Corporation
   2411 Garden Road
   Monterey, CA 93940
   Reference CRO #_____.

International Customers:

Equipment repair is handled by your local Pro-Log distributor. If you need to contact Pro-Log, the factory can be reached at any time by TWX at 910-360-7082.

LIMITED WARRANTY:

Seller warrants that the articles furnished hereunder are free from defects in material and workmanship and perform to applicable, published Pro-Log specifications for two years from date of shipment. This warranty is in lieu of any other warranty expressed or implied. In no event will Seller be liable for special or consequential damages as a result of any alleged breach of this warranty provision. This liability of Seller hereunder shall be limited to replacing or repairing, at its option, any defective units which are returned F.O.B. Seller's plant. Equipment or parts which have been subject to abuse, misuse, accident, alteration, neglect, unauthorized repair, or installation are not covered by warranty. Seller shall have the right of final determination as to the existence and cause of defect. As to items repaired or replaced, the warranty shall continue in effect for the remainder of the warranty period, or for ninety (90) days following date of shipment by Seller of the repaired or replaced part whichever period is longer. No liability is assumed for expendable items such as lamps and fuses. No warranty is made with respect to custom equipment or products produced to buyer's specifications except as specifically stated in writing by Seller and contained in the contract.

Cut Here

FOLD

FOLD

PRO-LOG® CORPORATION

Yes, I'm interested in Pro-Log.
Please send me more information about your:

- ☐ 7000 Series STD BUS Cards
- ☐ STD BUS Packaging Solutions
- ☐ STD BUS Subsystem
- ☐ STD BUS Systems
- ☐ Designer Software
- ☐ Design Courses and Seminars
- ☐ Programmer Product Line
- ☐ Other _____

- ☐ I am currently using Pro-Log equipment.
- ☐ Add my name to your mailing list.
- ☐ Please have a Sales Representative contact me.

Reader Comments: _____

Name _____ City _____
Company _____ State _____ Zip _____
Address _____ Telephone _____

PLACE
POSTAGE
HERE

PRO-LOG®
CORPORATION
2411 GARDEN ROAD
MONTEREY, CA 93940

Attn: Literature Department

Notes

Notes